AD-A078 781 CALIFORNIA UNIV DAVIS SIGNAL AND IMAGE PROCESSING LAB F/G 9/4
APPLICATIONS OF STOCHASTIC MODELS FOR IMAGE DATA COMPRESSION.(U)
SEP 79 S WANG , A K JAIN DAAG29-78-G-0206

UNCLASSIFIED ARO-16222.3-EL NL

1 OF 3
AD-
A078781

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 16222.3-EL	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Applications of Stochastic Models for Image Data Compresson		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Shenq-Huey Wang Anil K. Jain		8. CONTRACT OR GRANT NUMBER(s) DAAG29 78 G 0206
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California Davis, California 95616		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office P. O. Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE September 1979
		13. NUMBER OF PAGES 237
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

LEVEL II

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DDC RECEIVED DEC 27 1979 E

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

data compression
coding
images
mathematical models
stochastic processes

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Intraframe image data compression systems are analyzed in this thesis using stochastic modeling concepts. The formulations of stochastic image models are obtained from different classes of partial differential equations. Their application to the coding problem shows the connection between predictive, hybrid and transform coding schemes. The resulting coding schemes are evaluated in terms of tangible system terms such as signal to noise ratio, mean square error and rate distortion curve.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

APPLICATIONS OF STOCHASTIC MODELS FOR IMAGE DATA COMPRESSON

Technical Report

by

Shenq-Huey Wang

Anil K. Jain

Signal & Image Processing Laboratory

Department of Electrical Engineering

University of California

Davis, California 95616

September, 1979

(12)

6 APPLICATIONS OF STOCHASTIC MODELS FOR IMAGE DATA COMPRESSON.

9 Technical Report

by

10 Shenq-Huey Wang*

Anil K. Jain

Signal & Image Processing Laboratory

Department of Electrical Engineering

University of California

Davis, California 95616

15 DAAG29-78-G-Ø2Ø6

11 Sep[redacted]79

12 256

18 ARO

17 16222.3-EL

---

*The author was with the Signal & Image Processing Laboratory, Department of Electrical Engineering, University of California, Davis, California 95616. He is now with the Picker Corporation, Mayfield Village, Ohio 44143.

411 110

mt

The research work reported in this dissertation has been supported in part by the Naval Ocean Systems Center, San Diego, California, under contract N6600-1-78-C015MJE, and in part by the U.S. Army Research Office, Durham, North Carolina, under grant DAAG29-78-G0206.

TABLE OF CONTENTS

ABSTRACT

Intraframe image data compression systems are analyzed in this thesis using stochastic modeling concepts. The formulations of stochastic image models are obtained from different classes of partial differential equations. Their application to the coding problem shows the connection between predictive, hybrid and transform coding schemes. The resulting coding schemes are evaluated in terms of tangible system terms such as signal to noise ratio, mean square error and rate distortion curve.

Most of the existed intraframe coding schemes are either ad hoc or lack a complete mathematical basis. By paying attention to the boundary points of an image random field, it is shown that the remaining image random field can be fairly described by stochastic image models which in turn can be processed via well-defined fast transform algorithms (Sine and Cosine transforms are used here). The eigenvalues associated with these transform algorithms are important for coding application. To properly handle a nonstationary image random field, a new method based on representing an image as a composition of two sources, viz., stochastic and deterministic, is suggested. The stochastic part is dealt with by known statistical properties of stochastic image models and the deterministic part representing features such as edges, is obtained as the residual after the stochastic part is removed. Two simple adaptive schemes, i.e., Adaptive Variance Estimation and Adaptive Classification, are considered for updating the model parameters with the variations in image statistics. The computational complexity is increased only marginally, but the improvement in the reconstructed image quality is substantial.

Simulation studies are performed on three image data consisting of Girl, Earth and Chemical Plant images. The Results indicate that the proposed schemes can be practically implemented and will reproduce a good image at comparable signal to noise ratio. Hybrid coding of noisy image and sensitivity of human viewer to the sharp features are also partially addressed.

ACKNOWLEDGMENTS

I wish to express my gratefulness and appreciation to my thesis advisor, Professor Anil K. Jain. I cannot thank him adequately for the guidance, encouragement and support which he has so generously given me during the course of this research.

I would like to thank Dr. Peter D. Scott and Dr. Demetrios G. Lainiotis for graciously agreeing to serve on my Ph.D. program committee, and Dr. Gary Ford of the University of California, Davis, for serving as the outside reader.

Valuable discussions and assistance provided by Mr. Jaswant R. Jain and Mr. Surendra Ranganath, both Ph.D. students under Professor Jain's supervision, are also greatly appreciated.

Finally, a special note of thanks goes to Ms. Nora Rogers for the difficult job of typing the manuscript and putting extra hours on weekends to meet my schedule.

LIST OF FIGURES

* These figures contain images

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 The Problem

Digital image processing has attracted attention and found applications in diverse fields such as transmission of satellite pictures, enhancement of biomedical images, radar, seismic signal communication and many others. Generally speaking, digital image processing deals with two-dimensional data which is first sampled in spatial coordinates and then quantized in brightness. For typical images (used here), 256 × 256 image samples are taken and the brightness is quantized to 2^8 levels. This amounts to an equivalent data rate of $256 \times 256 \times 8 = 5.24 \times 10^5$ bits per image. Based on the current telemetry channel capacity of the order of 16.2 kbits/second [10], it would take about 32 seconds to transmit 5.24×10^5 bits. For many other data sources, such as satellite and biomedical image sources, the data rate are even higher. The large quantities of image data generated for real-time digital processing make it mandatory to consider data compression techniques. Data compression refers to representing an image by as small a number of bits as possible at an acceptable fidelity level.

Most image data are highly redundant, so that compression can be achieved by removing this redundancy before transmission over a communication channel. Data compression schemes for two- and three-dimensional images have been studied by many researchers. One particular class of methods is called statistical data compression where a class of images is characterized by a stochastic process which is described in terms of a mean and covariance function represented by a mathematical model.

Franks [12] has suggested a model for spectral density of two-dimensional video signals from which a separable covariance function can be developed. Nahi et al [48,49] and Habibi [20] and others have used this separable function model for processing images. This model, although mathematically convenient, is not very representative for many images. Jain [31] has proposed an approach where the image models are obtained by a finite difference approximation of second order partial differential equations (PDEs). This approach offers a rich mathematical basis for formulating and solving many important problems in image processing.

In this research, there are several objectives to be achieved. The primary objective is to further investigate the two-dimensional image models developed by Jain [31] for intraframe image data compression. The use of image models, i.e., *causal*, *semicausal and noncausal*, shows the connection between predictive, hybrid and transform coding schemes. From the choice of an appropriate image model, one would be able to determine an efficient algorithm for data compression. The second objective is to extend these modeling concepts to images which may be viewed as a composition of a stationary field and a deterministic field (e.g., edges which may vary with different images). Therefore, a coding technique can be designed to the sharp brightness changes and also to preserve the nice property of conventional transform coding. This technique is called feature transform coding.

1.2 Background Material

There are several important data compression techniques or concepts relevant to this research. They are briefly stated as follows:

1.2.1 Differential Pulse Code Modulation (DPCM)

The basic principle of a DPCM system is to quantize and encode the changes between successive samples, rather than the instantaneous sample values. It was first proposed by Cutler [9] in 1952 that a reduction in quantization noise for the same bit rate could be obtained by taking advantage of the redundancy or correlation between samples. Such correlation exists in practically all signals. It can be shown that the variance of the difference signals is considerably smaller than the variance of the original signals unless there is no correlation between successive samples. Harrison [23] showed the DPCM concept through the use of prediction theory. Later work published by O'Neal [50] showed a design procedure for DPCM and its application to television signals. His results gave some verification of modeling video signals by a first order Markov process. Other work related to the design of DPCM systems may be found in [10,45,68]. In our work we will use DPCM schemes in conjunction with transform coding and show their origin in some of the image models.

1.2.2 Transform Coding

An ideal transform coding system consists of a unitary, linear mapping of a block of correlated samples into a set of statistically independent coefficients. Data compression is then accomplished by

sorting and quantizing the transform coefficients according to their information capacity.

Andrews and Pratt [5] introduced the concept of transform coding of images via Fourier transform in 1968, which led to the investigation of utilizing various unitary transforms for image processing [1,3,4,21,26,55,70] Among the candidates for transform coding, the Karhunen-Loeve transform provides minimum mean square error for any fixed rate. Common image transforms are discrete Fourier, Hadamard, Cosine, Sine, Slant, etc. (refer to Chapter 4).

In our work we will show that a particular class of image models, namely noncausal models, lead naturally to transform coding algorithms. It will be shown that the choice of transform and the coder design parameters are determined via the model parameters.

1.2.3 Hybrid (Transform/DPCM) Coding

Hybrid coding is a technique which combines the advantages of transform coding and DPCM coding, which was first proposed by Habibi [17] for two-dimensional images modeled by a separable covariance function. Subsequently, Roese et al [58] extended this technique for interframe coding of images. Recursive filtering of two- and three-dimensional images based on hybrid coding concept was proposed by Jain and Angel [33]. Other studies of hybrid coding schemes can be found in [18].

In our work, semicausal image models lead naturally to the hybrid coding algorithms and the adaptive and nonadaptive schemes of implementing those algorithms are presented in detail.

1.2.4 Contour Coding and Edge Extraction

Since a significant amount of visual information in an image lies in the edges, it is reasonable to consider coding schemes where the edges are treated more adequately so that the processed image would appear acceptable to a human observer.

In 1958 Schreiber and Graham [15,62-65] introduced an edge coding technique called synthetic highs, in which a scanned video signal is segmented into its high and low frequency components and they are transmitted separately. The high frequency component essentially contains the edge information. At the receiver, the high frequency component is synthesized from the addresses of the image edges and combined with low frequency component to generate the original video signal. Yan and Sakrison [74] proposed a similar scheme where an image is represented by two sources, texture and edge. Image quality can be improved by simply preprocessing with an appropriate method before transmission and postprocessing with its inverse after reception.

Use of image models for segmenting high and low frequency components will be shown in this research where several simulation results are illustrated to show the effectiveness of the new process.

1.3 Organization

Having outlined the background material and the importance of data compression, we can now describe the organization of the thesis. Chapter two provides a general discussion of image models and their properties. The efficiency of these models is measured by comparing their data compression ability in terms of basis restriction errors (for definition see Chapter two). Chapter three is devoted to the development of

Hybrid (Transform/DPCM) coding algorithms. It is shown that the semicausal class of models leads to such algorithms. Four different semicausal models have been considered. Adaptive schemes are designed by simply adapting the model parameters to cope with the changes in image statistics. The effect of a noisy transmission channel on the Hybrid coding system is also studied. It is shown that the semicausal image models may be used to develop a Hybrid coding system for compression of noisy images. In Chapter four, we show how the noncausal image models may be used to design transform coders. Adaptive and nonadaptive algorithms are considered. In Chapter five, we extend the algorithms of Chapter four to represent images as a composition of two sources. One of the sources is a stationary component and the other is a nonstationary or deterministic component. Examples of computer simulation of all the algorithms on image data are given and comparisons are made.

Finally, the Appendices contain the derivation and proofs that relate to various topics throughout the thesis. In particular, Appendix A provides useful tabulated data for various quantizers and serves as a good reference for engineers interested in designing quantizers for image processing applications.

CHAPTER TWO

DISCRETE GAUSSIAN RANDOM FIELDS AND STOCHASTIC IMAGE MODELS

2.1 Introduction

In discussing image representations, we think of characterizing images as two-dimensional random fields. If D is a constant coefficient difference equation operator, we consider representations of the form

$$D[u_{ij}] = \varepsilon_{ij} \tag{2.1-1}$$

where $\{u_{ij}\}$ denotes the array of image pixels, $\{\varepsilon_{ij}\}$ is a two dimensional white noise process or a moving average field.

In this chapter, we consider causal, semicausal and noncausal representations which are of the form of (2.1-1). These representations, as we will see in Chapters 3, 4 and 5, will aid in developing useful algorithms for many image coding problems. Here we will study data compression efficiency of these models measured by the basis restriction error.

2.2 Causal, Semicausal and Noncausal Estimates

Let $\{u_{i,j}\}$ represent a zero mean, stationary two-dimensional discrete Gaussian random field whose covariance function is defined as

$$E\, u_{i,j} u_{i+k,j+\ell} = r(k,\ell) . \tag{2.2-1}$$

Let $\bar{u}_{i,j}$ denote an estimate of the random variable $u_{i,j}$. We consider the following three types of estimates.

2.2.1 Causal Estimate

Suppose the elements of the random field $\{u_{i,j}\}$ are arranged in any

desired, one-dimensional, ordered sequence. Then $\bar{u}_{i,j}$ is a causal estimate of $u_{i,j}$ if it depends only on the elements that occur before the element $u_{i,j}$. A common example occurs when the image is scanned column by column and $\bar{u}_{i,j}$ is a linear estimate based on all the elements scanned before arriving at (i,j), i.e.,

$$\bar{u}_{i,j} = \sum_{m,n \in S} a(m,n) u_{i-m,j-n} \qquad (2.2.1\text{-}1)$$

where $S = \{m,n: n > 0, \forall m\} \cup \{m,n: n = 0, m > 0\}$. Fig. 2.1(a) shows the set S for causal estimate at (i,j).

2.2.2 Semicausal Estimate

If the estimate $\bar{u}_{i,j}$ is causal in one of the coordinates and noncausal in the other, it is called a semicausal estimate. For example, a linear semicausal estimate which is causal in "j" and noncausal in "i" would be of the form

$$\bar{u}_{i,j} = \sum_{m,n \in S} a(m,n) u_{i-m,j-n} \qquad (2.2.2\text{-}1)$$

where $S = \{m,n: n > 0, \forall m\} \cup \{m,n: n = 0, \forall m \neq 0\}$ and is shown in Fig. 2.1(b).

2.2.3 Noncausal Estimate

The quantity $\bar{u}_{i,j}$ is a linear noncausal estimate of $u_{i,j}$ if it can be written as a linear function of possibly all the variables in the random field, except $u_{i,j}$ itself. For example, a linear noncausal estimate would be of the type

$$\bar{u}_{i,j} = \sum_{m,n \in S} a(m,n) u_{i-m,j-n} \tag{2.2.3-1}$$

where $S = \{m,n: \forall(m,n) \neq (i,j)\}$ and is shown in Fig. 2.1(c). Note that $\bar{u}_{i,j}$ contains terms from all the four quadrants about the point (i,j).

2.3 Stochastic Representations of Gaussian Random Field

Let $\bar{u}_{i,j}$ be an arbitrary estimate of $u_{i,j}$, then a stochastic representation of the Gaussian random field $\{u_{i,j}\}$ is defined as

$$u_{i,j} = \bar{u}_{i,j} + \varepsilon_{i,j} \tag{2.3-1}$$

where $\{\varepsilon_{i,j}\}$ is another random field.

There are two types of representations which are of interest to be considered here. These are as follows:

1) Minimum variance representation.

2) White noise driven representation.

2.3.1 Minimum Variance Representation

A minimum variance estimate is one which minimizes the mean square error

$$e_{i,j} = E(u_{i,j} - \bar{u}_{i,j})^2 \tag{2.3.1-1}$$

at each (i,j). For Gaussian random field, the minimum variance estimate would be linear. The coefficients a(m,n) in (2.2.1-1), (2.2.2-1) or (2.2.3-1) could be determined from the covariance function $r(k,\ell)$ and the orthogonality relation

$$E(u_{i,j} - \overline{u}_{i,j})u_{p,q} = E\left[u_{i,j} - \sum_{m,n\in S} a(m,n)u_{i-m,j-n}\right]u_{p,q} = \beta^2_{i,j}\delta_{i,p}\delta_{j,q}$$

for all $p,q \in \hat{S}$. $\hat{S} = S \cup (i,j)$ (2.3.1-2)

where S depends on whether $\overline{u}_{i,j}$ is causal, semicausal or noncausal, and $\beta^2_{i,j}$ is defined to be the minimized value of $e_{i,j}$ of (2.3.1-1).

For minimum variance representation, $\overline{u}_{i,j}$ is chosen to be a minimum variance estimate. This representation could be causal, semicausal or noncausal in its spatial structure.

2.3.2 White Noise Driven Representation

A white noise random field $\{x_{i,j}\}$ is defined as one whose elements are mutually uncorrelated, i.e.,

$$Ex_{i,j}x_{k,\ell} = \beta^2_{i,j}\delta_{i,k}\delta_{j,\ell}. \quad (2.3.2\text{-}1)$$

For white noise driven representation, $\{\varepsilon_{i,j}\}$ in (2.3-1) is chosen to be a white noise random field.

2.4 Spectral Density Functions of the Gaussian Random Field Represented by Linear Models

Consider the stationary Gaussian random field $\{u_{i,j}\}$ defined on the infinite plane, $-\infty < i,j < \infty$. The two-dimensional z-transform* of the covariance function, called the covariance generating function, is defined as

---

* Two-dimensional z-transform is defined as

$$F(z_1,z_2) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i,j)z_1^{-i}z_2^{-j}.$$

$$S_u(z_1,z_2) = \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} r(k,\ell) z_1^{-k} z_2^{-k} \tag{2.4-1}$$

Substituting $e^{j\omega_1}$ and $e^{j\omega_2}$ for z_1 and z_2, respectively, we have

$$S_u(\omega_1,\omega_2) = S_u(z_1,z_2) \Big|_{z_1 = e^{j\omega_1},\ z_2 = e^{j\omega_2}} . \tag{2.4-2}$$

This becomes the spectral density function (SDF) of $\{u_{i,j}\}$. For simplicity, we refer to $S_u(z_1, z_2)$ as the spectral density function of the Gaussian random field $\{u_{i,j}\}$ at various places in the following chapters without further notification.

The discrete operator of (2.1-1) can be written as

$$D(z_1, z_2) = 1 - \sum_{m,n\varepsilon S} a(m,n) z_1^{-m} z_2^{-n} \tag{2.4-3}$$

If $S_\varepsilon(z_1,z_2)$ is the spectral density function of $\{\varepsilon_{i,j}\}$, then the spectral density function of $\{u_{i,j}\}$ is given by

$$S_u(z_1,z_2) = \frac{S_\varepsilon(z_1,z_2)}{D(z_1,z_2)\,D(z_1^{-1},z_2^{-1})} . \tag{2.4-4}$$

The covariance function of $\{u_{i,j}\}$ is obtained simply by the inverse Fourier transform of $S_u(z_1, z_2)$

$$R(k,\ell) = \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{2\pi} S_u(\omega_1,\omega_2) \exp(j\omega_1 k + j\omega_2 \ell)\, d\omega_1 d\omega_2 \tag{2.4-5}$$

2.5 Image Representations

Now we consider different types of random field representations as candidates for image modeling.

2.5.1 Causal Representations

The general form of causal representations of an image field is given by

$$u_{i,j} = \bar{u}_{i,j} + \varepsilon_{i,j} \tag{2.3-1}$$

where $\bar{u}_{i,j}$ is the causal estimate of $u_{i,j}$ defined in (2.2.1-1).

As an example, one causal representation is of the type

$$u_{i,j} = a_1 u_{i-1,j} + a_2 u_{i,j-1} + a_3 u_{i-1,j-1} + \varepsilon_{i,j} \tag{2.5.1-1}$$

whose spatial structure is depicted in Table 2.1 and it is obvious that $u_{i,j}$ is causally related with three points which occurred at the left upper quadrant. It has been shown that this equation and other similar causal representations can be obtained by discrete approximation of hyperbolic partial differentail equations [34].

Using the spatial structure and measured covariances of a given image field, one may identify the coefficients a_1, a_2, a_3 and the random field $\{\varepsilon_{i,j}\}$ via least square regression techniques. For the covariance function

$$r(k,\ell) = Eu_{i,j}u_{i+k,j+\ell} = \rho_1^{|k|}\rho_2^{|\ell|} \tag{2.5.1-2}$$

where ρ_1 and ρ_2 are the vertical and horizontal correlations respectively.

The model of (2.5.1-1) is its exact realization when

$$a_1 = \rho_1 , \; a_2 = \rho_2 , \; a_3 = -\rho_1\rho_2 \tag{2.5.1-3}$$

and $\{\varepsilon_{i,j}\}$ is a white noise field with zero mean and covariance function

$$E\, \varepsilon_{i,j}\varepsilon_{i+k,j+\ell} = \beta^2\delta_{k,0}\delta_{\ell,0} , \quad \beta^2 = (1-\rho_1^2)(1-\rho_2^2). \tag{2.5.1-4}$$

The corresponding discrete operator and spectral density function of (2.5.1-1) subject to (2.5.1-3) are given as follows:

$$D(z_1,z_2) = (1-\rho_1 z_1^{-1})(1-\rho_2 z_2^{-1}) \tag{2.5.1-5a}$$

$$S_\varepsilon(z_1,z_2) = \beta^2 = (1-\rho_1^2)(1-\rho_2^2) \tag{2.5.1-5b}$$

$$S_u(z_1,z_2) = \frac{\beta^2}{(1-\rho_1 z_1)(1-\rho_1 z_1^{-1})(1-\rho_2 z_2)(1-\rho_2 z_2^{-1})} . \tag{2.5.1-5c}$$

In image processing literature, this model has been used for differential PCM and filtering of images [34].

2.5.2 Semicausal Representations

A semicausal representation is causal in one of the image coordinates (say, j) and is noncausal in the other (say, i). For example,

$$u_{i,j} = \alpha(u_{i-1,j} + u_{i+1,j}) + \gamma u_{i,j-1} + \varepsilon_{i,j} \tag{2.5.2-1}$$

where $|\alpha| < \frac{1}{2}$, $|\gamma| < 1$ and $|2\alpha + \gamma| < 1$. (2.5.2-1a)

This is a semicausal representation of the Gaussian random field. Conditions of (2.5.2-1a) insure the stability of (2.5.2-1). The covariance function generated by this equation depends on the covariance

function chosen for the random field $\{\varepsilon_{i,j}\}$. For example, when $\{\varepsilon_{i,j}\}$ is a white noise field, i.e.,

$$r_\varepsilon(k,\ell) = E\varepsilon_{i,j}\varepsilon_{i+k,j+\ell} = \beta^2\delta_{k,0}\delta_{\ell,0}. \qquad (2.5.2\text{-}2)$$

The spectral density function of $\{u_{i,j}\}$ is then given by

$$S_u(z_1,z_2) = \frac{\beta^2}{(1-\alpha z_1-\alpha z_1^{-1} - \gamma z_2)(1-\alpha z_1^{-1}-\alpha z_1-\gamma z_2^{-1})}. \qquad (2.5.2\text{-}3)$$

This white noise driven representation will be called the <u>white noise driven SC1 model</u>.

As a generalization, consider the case when $\{\varepsilon_{i,j}\}$ has its covariance function given by

$$r_\varepsilon(k,\ell) = E\varepsilon_{i,j}\varepsilon_{i+k,j+\ell} = \beta^2\delta_{\ell,0}(\delta_{k,0}-\alpha_1\delta_{k-1,0}-\alpha_1\delta_{k+1,0}). \qquad (2.5.2\text{-}4)$$

The corresponding SDFs are given by

$$S_\varepsilon(z_1,z_2) = \beta^2[1-\alpha_1(z_1+z_1^{-1})] \qquad (2.5.2\text{-}5a)$$

$$S_u(z_1,z_2) = \frac{S_\varepsilon(z_1,z_2)}{(1-\alpha z_1-\alpha z_1^{-1}-\gamma z_2)(1-\alpha z_1^{-1}-\alpha z_1-\gamma z_2^{-1})}. \qquad (2.5.2\text{-}5b)$$

For $\alpha_1 = 0$, (2.5.2-3) results, we have white noise driven model.

For $\alpha_1 = \alpha$, the quantity [see (2.5.2-1)]

$$u^*_{i,j} = \alpha(u_{i-1,j} + u_{i+1,j}) + \gamma u_{i,j-1} \qquad (2.5.2\text{-}6)$$

becomes the best linear mean square estimate of $u_{i,j}$ given k,ℓ belonging to

$$S = \{k,\ell\colon \ell < j,\ \forall k\} \cup \{k,\ell\colon \ell = j,\ \forall k \neq i\}.$$

This follows by observing that the orthogonality relation

$$E u^*_{i,j}(u_{i,j} - u^*_{i,j}) = E u^*_{i,j}\varepsilon_{i,j} = 0, \ \forall\, i,j \qquad (2.5.2\text{-}7)$$

is satisfied when $\alpha_1 = \alpha$.

Then (2.5.2-1) becomes a minimum variance SC1 representation, since $E(u_{i,j} - u^*_{i,j})^2$ is minimum over the set of lattice points of S. The spectral density function of the minimum variance semicausal representation is obtained by setting $\alpha_1 = \alpha$ in (2.5.2-5).

It has been shown by Jain [28] that the separable covariance function [see (2.5.1-2)] has a semicausal minimum variance realization given by

$$u_{i,j} = \alpha_1(u_{i-1,j} + u_{i+1,j}) - \rho_2\alpha_1(u_{i-1,j-1} + u_{i+1,j-1}) + \rho_2 u_{i,j-1}\varepsilon_{i,j} \qquad (2.5.2\text{-}8)$$

where $\{\varepsilon_{i,j}\}$ is chosen to have its covariance given as

$$r_\varepsilon(k,\ell) = \frac{(1-\rho_1^2)(1-\rho_2^2)}{(1+\rho_1^2)}\,\delta_{\ell,0}(-\alpha_1\delta_{k,-1} - \alpha_1\delta_{k,1} + \delta_{k,0}) \qquad (2.5.2\text{-}9)$$

and

$$\alpha_1 = \frac{\rho_1}{1+\rho_1^2}$$

with corresponding SDF

$$S_\varepsilon(z_1,z_2) = \frac{(1-\rho_1^2)(1-\rho_2^2)}{(1+\rho_1^2)}\,(1-\alpha_1 z_1 - \alpha_1 z_1^{-1}) \,. \qquad (2.5.2\text{-}10)$$

This will be called the minimum variance SC2 representation. Interestingly, the spectral density function $S_u(z_1,z_2)$ of this minimum variance SC2 representation and that of the white noise driven C1 representation are identical. For more general minimum variance semicausal models, see [34].

The semicausal representations will be shown to lead some useful hybrid algorithms for image coding. A hybrid algorithm is a recursive algorithm obtained after transforming each row (or column) of the image by a unitary transformation. Note that above mentioned semicausal representations can also be obtained by discrete difference approximation of parabolic partial differential equations [34].

2.5.3 Noncausal Representations

The noncausal representations are boundary value problems in each coordinate and have correspondence with the elliptic class of partial differential equations (PDEs) [34]. For example, the noncausal representation (NC1)

$$u_{i,j} = \alpha(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) + \varepsilon_{i,j} \ , \ |\alpha| < \tfrac{1}{4} \qquad (2.5.3\text{-}1)$$

is a discrete approximation of a Poisson PDE.

In general, $\{\varepsilon_{i,j}\}$ may be chosen such that

$$r_\varepsilon(k,\ell) = E\varepsilon_{i,j}\varepsilon_{i+k,j+\ell} = \beta^2[\delta_{k,0}\delta_{\ell,0} - \alpha_1(\delta_{k,-1}+\delta_{k,1})\delta_{\ell,0} - \alpha_1(\delta_{\ell,-1}+\delta_{\ell,1})\delta_{k,0}] \ , \ |\alpha_1| < \tfrac{1}{4} \qquad (2.5.3\text{-}2)$$

The spectral density functions for $\{\varepsilon_{i,j}\}$ and $\{u_{i,j}\}$ are then obtained as

$$S_\varepsilon(z_1,z_2) = \beta^2[1 - \alpha_1(z_1+z_1^{-1}+z_2+z_2^{-1})] \qquad (2.5.3\text{-}3)$$

$$S_u(z_1,z_2) = \frac{S_\varepsilon(z_1,z_2)}{[1 - \alpha(z_1+z_1^{-1}+z_2+z_2^{-1})]^2} \ . \qquad (2.5.3\text{-}4)$$

There are three types of NC1 models to be considered here.

a. $\alpha_1 = 0$. $\{\varepsilon_{i,j}\}$ becomes a white noise process and (2.5.3-1) is a <u>white noise driven NC1 representation</u>.

b. $\alpha_1 = \alpha$, we get the <u>minimum variance NC1 representation</u>. Note that in the spectral density function S_u a pole-zero cancellation occurs and the S_u becomes

$$S_u(z_1,z_2) = \frac{\beta^2}{[1 - \alpha(z_1+z_1^{-1}+z_2+z_2^{-1})]} . \qquad (2.5.3\text{-}5)$$

c. For $\alpha_1 \neq \alpha$, (2.5.3-1) may be considered as a noncausal autoregressive moving average (ARMA) model, denotes a <u>NC1 ARMA model</u>. According to Jain [34], model identification experiments have shown for several images (for example, Girl, Moon and Earth) that this model gives a better fit to the real data than the previous two models.

Another type of noncausal representation can be derived from a biharmonic PDE [34], and gives a 13-neighbor model

$$u_{i,j} = a(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) - a\alpha(u_{i-1,j-1} + u_{i-1,j+1}$$

$$+ u_{i+1,j-1} + u_{i+1,j+1}) - \frac{a\alpha}{2}(u_{i-2,j} + u_{i+2,j} + u_{i,j-2} + u_{i,j+2}) + \frac{\varepsilon_{i,j}}{(1+4\alpha^2)}$$

$$(2.5.3\text{-}6)$$

where $a = \dfrac{2\alpha}{(1+4\alpha^2)}$.

If covariances of $\{\varepsilon_{i,j}\}$ are taken as

$$r_\varepsilon(k,\ell) = E\varepsilon_{i,j}\varepsilon_{i+k,j+\ell} = \beta^2 \begin{cases} -a_1 & \text{, if } \ell=\pm1,\ k=0 \text{ or } \ell=0,\ k=\pm1 \\ a_1\alpha_1 & \text{, if } \ell=\pm1,\ k=\pm1 \\ \dfrac{a_1\alpha_1}{2} & \text{, if } \ell=\pm2,\ k=0 \text{ or } k=\pm2,\ \ell=0 \\ 1 & \text{, if } k=\ell=0 \\ 0 & \text{, otherwise} \end{cases}$$

(2.5.3-7)

where $|a_1| < \frac{2}{5}$, $|\alpha_1| < \frac{1}{4}$;

then, for $a_1 = a$, $\alpha_1 = \alpha$

Eqn. (2.5.3-6) becomes a <u>minimum variance NC2 representation</u>. Spectral density functions for this mode are given by

$$S_\varepsilon(z_1,z_2) = \beta^2[1 - \alpha_1(z_1+z_1^{-1}+z_2+z_2^{-1})]^2 \qquad (2.5.3\text{-}8)$$

$$S_u(z_1,z_2) = \frac{S_\varepsilon(z_1,z_2)}{[1 - \alpha(z_1+z_1^{-1}+z_2+z_2^{-1})]^4} . \qquad (2.5.3\text{-}9)$$

The noncausal representations discussed above often realize nearly isotropic $S_u(z_1, z_2)$.

Not all noncausal models give near-isotropic random fields. For example, consider the noncausal model (NC3)

$$\begin{aligned} u_{i,j} = {} & \alpha_1(u_{i-1,j} + u_{i+1,j}) + \alpha_2(u_{i,j-1} + u_{i,j+1}) \\ & - \alpha_1\alpha_2(u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1}) + \varepsilon_{i,j} . \end{aligned} \qquad (2.5.3\text{-}10)$$

If the covariances of $\{\varepsilon_{i,j}\}$ are given by

$$r_\varepsilon(k,\ell) = E\varepsilon_{i,j}\varepsilon_{i+k,j+\ell} = \beta^2 \begin{cases} 1 \;, & \text{if } k=\ell=0 \\ -\alpha_1 \;, & \text{if } k=\pm 1,\ \ell=0 \\ -\alpha_2 \;, & \text{if } k=0,\ \ell=\pm 1 \\ \alpha_1\alpha_2 \;, & \text{if } k=\pm 1,\ \ell=\pm 1 \\ 0 \;, & \text{otherwise} \end{cases} \tag{2.5.3-11}$$

where $\beta^2 = \dfrac{(1-\rho_1^2)(1-\rho_2^2)}{(1+\rho_1^2)(1+\rho_2^2)}$, $\alpha_1 = \dfrac{\rho_1}{(1+\rho_1^2)}$, $\alpha_2 = \dfrac{\rho_2}{(1+\rho_2^2)}$.

SDFs of (2.5.3-10) are given by

$$S_\varepsilon(z_1,z_2) = \frac{\beta^2}{[1-\alpha_1(z_1+z_1^{-1})][1-\alpha_1(z_2+z_2^{-1})]} \tag{2.5.3-12}$$

$$S_u(z_1,z_2) = \frac{(1-\rho_1^2)(1-\rho_2^2)}{(1-\rho_1z_1)(1-\rho_1z_1^{-1})(1-\rho_2z_2)(1-\rho_2z_2^{-1})} . \tag{2.5.3-13}$$

Notice that the form of $S_u(z_1z_2)$ is the same as the SDF of causal DPCM C1 model and semicausal SC2 model, so it generates the separable covariance model which is non-isotropic.

2.6 Basis Restriction Data Compression Efficiency of Image Models

The performance of the foregoing image models in image data compression can be analyzed by studying their basis restriction efficiency. The scheme is shown in Fig. 2.2. First, the original image $\{u_{i,j}\}$ is transformed by a unitary matrix A (see Chapter 4, the definition of transformation). Next, a two-dimensional sample selection function $\{w(i,j)\}$ (also called zonal filter), which at each (i,j) takes on the

value of zero or one, is applied to the transformed image $\{v_{i,j}\}$. Only those samples for which $\{w(i,j)=1\}$ are stored, zeros placed at other locations. This zonal filtered image $\{\hat{v}_{i,j}\}$ is inverse transformed by A^{-1} to give $\{\hat{u}_{i,j}\}$.

The basis restriction mean square error (B.R.M.S.E.) is defined as

$$\text{B.R.M.S.E.} = \sum_i \sum_j E\,(u_{i,j} - \hat{u}_{i,j})^2 \Big/ \sum_i \sum_j u_{ij}^2$$

$$= \sum_i \sum_j E\,(v_{i,j} - \hat{v}_{i,j})^2 \Big/ \sum_i \sum_j v_{ij}^2$$

$$= \sum_i \sum_j E\left[v_{i,j}(1 - w(i,j))\right]^2 \Big/ \sum_i \sum_j v_{ij}^2 \qquad (2.6\text{-}1)$$

where we have used the fact that A is a unitary transform. Associated with this error is the sample reduction ratio (S.R.R.)

$$\text{S.R.R.} = \frac{\text{Number of image samples}}{\text{Number of samples selected by zonal filter}} . \qquad (2.6\text{-}2)$$

This ratio is a measure of data compression.

2.6.1 Zonal Filter Design

The zonal filter $w(i,j)$ is designed as follows.

1. Let t_n be the number of samples to be selected for transmission for a given S.R.R. Clearly t_n = NM/S.R.R. for an M × N image.
2. Find the variances $\sigma^2_{m,n}$ of the transformed image samples $v_{m,n}$. If A is the DFT, and N is large, an estimate of $\sigma^2_{m,n}$ is

$$\sigma^2_{m,n} = \begin{cases} S_u\left(\frac{2\pi m}{M}, \frac{2\pi n}{N}\right), & 0 \le |m| \le \frac{M}{2},\ 0 \le |n| \le \frac{N}{2} \\ \sigma^2_{M-m,N-n}, & \text{otherwise} \end{cases}$$

3. Arrange $\sigma^2_{m,n}$ in a decreasing order. Then $w(m,n) = 1$ for the first t_n addresses (m,n) and $w(m,n) = 0$ otherwise.

Since $w(m,n)$ depend on $\sigma^2_{m,n}$ or the SDF, the zonal filter functions will be different for different image models.

2.6.2 Experimental Results

For easy reference, all the image representations (models) presented in Section 2.5 are summarized in Table 2.1. It is seen that various partial differential equation (PDE) operators are also listed. Table 2.2 shows model parameters used for three different images. These parameters were found by least square matching of the 16 × 16 image covariances with the model covariances.

Simple basis restriction error experiments have been performed to compare the various image models for the Girl and Earth images. We have used the discrete Fourier transform to be the unitary transform operator. Since the image models differ in their SDFs, in the transform domain, the variances of the samples $v_{i,j}$ will have different distributions. Figs. 2.3 - 2.5 show the zonal filters at various sample reduction ratios for causal, semicausal and noncausal models. Table 2.3 lists the results of basis restriction mean square error versus image block size for the Girl image. Table 2.5 lists the corresponding results for the Earth image. Figs. 2.6 - 2.7 show the plots of basis restriction errors versus

sample reduction ratios. In general, basis restriction error of the noncausal model is lower for most sample reduction ratios (indicating its superior fitting of the image data than the other two types of models. For very large sample reduction ratios, since few transform domain image pixels are transmitted, all the models select the same pixels (see Figs. 2.6 - 2.7)).

The above conclusion could also be drawn from Figs. 2.8 and 2.9, which show the Basis Restriction Data Compression of Girl and Earth images using a 16 × 16 image block size. Clearly, noncausal models give the best results followed by semicausal models.

2.7 Summary and Conclusions

In summary, several image representations have been considered. They originate from finite difference approximations of certain PDEs. These models have been compared on the basis of their data compression efficiency based on a basis restriction error criterion.

The following conclusions are made.

1. The models C1, SC2 and NC3 all generate random fields whose covariance functions are the same as the separable model [see (2.5.1-2)].
2. For $\alpha_1 = \alpha$, the spectral density of NC2 is the same as that of NC1 for $\alpha_1 = 0$. However, NC2 is a minimum variance noncausal representation, whereas NC1 is a white noise driven model.
3. In a data compression application, the noncausal models give the best performance. A similar conclusion has been made in image filtering applications [34].

4. Figs. 2.3 - 2.5 could be interpreted as the spectral density function indicators. Different types of image models would achieve different shapes of spectral density functions. Figs. 2.3 - 2.5 show the contours of approximately equal spectrum values.

5. Higher order models may be needed to represent image SDFs more accurately. Identification of such models remains an open problem.

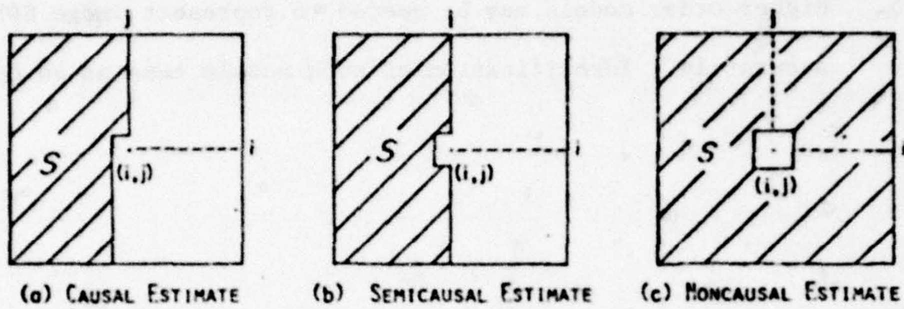


Figure 2.1 The Region S for Estimating (i,j)

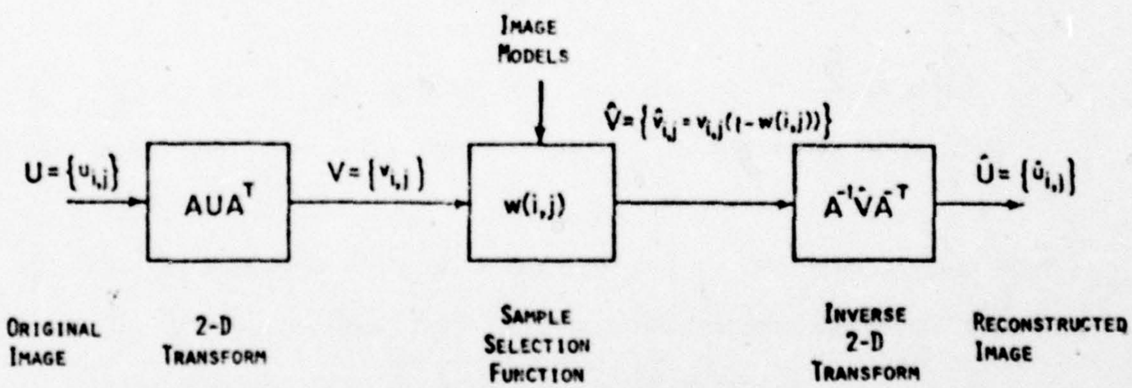


Figure 2.2 Basis Restriction Efficiency Implementation

```
00000000000001111111100000000000
00000000000001111111100000000000
00000000000001111111100000000000
00000000000011111111100000000000
00000000000011111111100000000000
00000000000111111111100000000000
00000000000111111111100000000000
00000000001111111111100000000000
00000000001111111111110000000000
00000000011111111111111000000000
00000000111111111111111100000000
00000011111111111111111111000000
00001111111111111111111111111000
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
00001111111111111111111111111000
00000001111111111111111110000000
00000000011111111111111100000000
00000000011111111111110000000000
00000000001111111111100000000000
00000000000111111111100000000000
00000000000011111111100000000000
00000000000011111111100000000000
00000000000001111111000000000000
00000000000001111111000000000000
00000000000001111111000000000000
00000000000001111111000000000000
```

(a) 2:1 Sample Reduction

```
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000111110000000000000
00000000000000111110000000000000
00000000000000111110000000000000
00000000000000111111100000000000
00000000000001111111110000000000
00000000000011111111111000000000
00000000111111111111111100000000
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
00000000111111111111111110000000
00000000000011111111100000000000
00000000000001111111000000000000
00000000000000111111000000000000
00000000000000111110000000000000
00000000000000111110000000000000
00000000000000111110000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
```

(b) 4:1 Sample Reduction

```
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000111110000000000000
00000000000001111111000000000000
00000000011111111111111100000000
11111111111111111111111111111111
00000000011111111111111100000000
00000000000000111111000000000000
00000000000000111110000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000011100000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
```

(c) 8:1 Sample Reduction

```
00000000000000000000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000011100000000000000
01111111111111111111111111111111
00000000000000011100000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
00000000000000001000000000000000
```

(d) 16:1 Sample Reduction

Figure 2.3 Causal Model (C1) Fourier Transform Domain Model Zonal Patterns for a 32 × 32 Image Block Size with Various Sample Reduction Ratios.

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000111111111111111110000000
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
00000000011111111111111100000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

(a) 2:1 Sample Reduction

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000001111111000000000000
00000000011111111111111100000000
00011111111111111111111111111100
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
00011111111111111111111111111100
00000000011111111111111100000000
00000000000000111110000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

(b) 4:1 Sample Reduction

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000011100000000000000
00000000000111111111110000000000
00000000111111111111111110000000
00000011111111111111111111100000
00001111111111111111111111110000
00000011111111111111111111100000
00000000111111111111111110000000
00000000000111111111110000000000
00000000000000011100000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

(c) 8:1 Sample Reduction

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000111110000000000000
00000000000011111111100000000000
00000000001111111111110000000000
00000000001111111111111000000000
00000000000111111111110000000000
00000000000001111111100000000000
00000000000000111110000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

(d) 16:1 Sample Reduction

Figure 2.4 Semicausal Model (SC1) Fourier Transform Domain Model Zonal Patterns for a 32 × 32 Image Block Size with Various Sample Reduction Ratios.

```
00000000000000000000000000000000
00000000000000001000000000000000
00000000000000111100000000000000
00000000000001111110000000000000
00000000000011111111100000000000
00000000000111111111111000000000
00000000001111111111111000000000
00000000011111111111111100000000
00000000111111111111111110000000
00000001111111111111111111000000
00000011111111111111111111100000
00000111111111111111111111110000
00001111111111111111111111111000
00011111111111111111111111111100
00111111111111111111111111111100
00111111111111111111111111111110
01111111111111111111111111111111
00111111111111111111111111111110
00011111111111111111111111111100
00001111111111111111111111111000
00001111111111111111111111110000
00000111111111111111111111100000
00000011111111111111111111000000
00000001111111111111111110000000
00000000111111111111111100000000
00000000011111111111111000000000
00000000001111111111110000000000
00000000000111111111100000000000
00000000000011111111000000000000
00000000000000111110000000000000
00000000000000011100000000000000
00000000000000001000000000000000
```

(a) 2:1 Sample Reduction

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000011110000000000000
00000000000001111111000000000000
00000000000011111111100000000000
00000000001111111111111000000000
00000000011111111111111100000000
00000000011111111111111100000000
00000000111111111111111110000000
00000001111111111111111110000000
00000001111111111111111111000000
00000001111111111111111111000000
00000001111111111111111111000000
00000001111111111111111110000000
00000000111111111111111110000000
00000000011111111111111100000000
00000000001111111111111100000000
00000000000111111111111000000000
00000000000111111111110000000000
00000000000001111111000000000000
00000000000000011100000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

(b) 4:1 Sample Reduction

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000111110000000000000
00000000000001111111000000000000
00000000000011111111100000000000
00000000000111111111110000000000
00000000001111111111111000000000
00000000001111111111111000000000
00000000001111111111111000000000
00000000001111111111111000000000
00000000000111111111110000000000
00000000000011111111100000000000
00000000000001111111000000000000
00000000000000111110000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

(c) 8:1 Sample Reduction

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000111110000000000000
00000000000001111110000000000000
00000000000011111111000000000000
00000000000011111111100000000000
00000000000011111111100000000000
00000000000011111111100000000000
00000000000011111111000000000000
00000000000001111110000000000000
00000000000000111000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

(d) 16:1 Sample Reduction

Figure 2.5 Noncausal Model (NC1 or NC2-A or NC2-B) Fourier Transform Domain Model Zonal Patterns for a 32 × 32 Image Block Size with Various Sample Reduction Ratios.

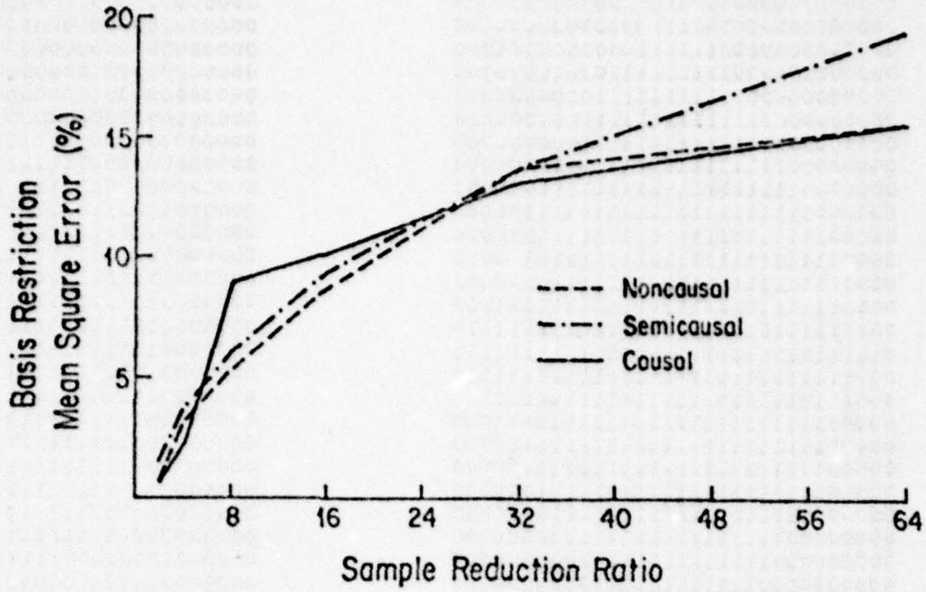


Figure 2.6 Basis Restriction Mean Square Error versus Sample Ratio for the Fourier Transform Model Zonal Sampling of a 256 × 256 Girl Image with 16 × 16 Image Block Size

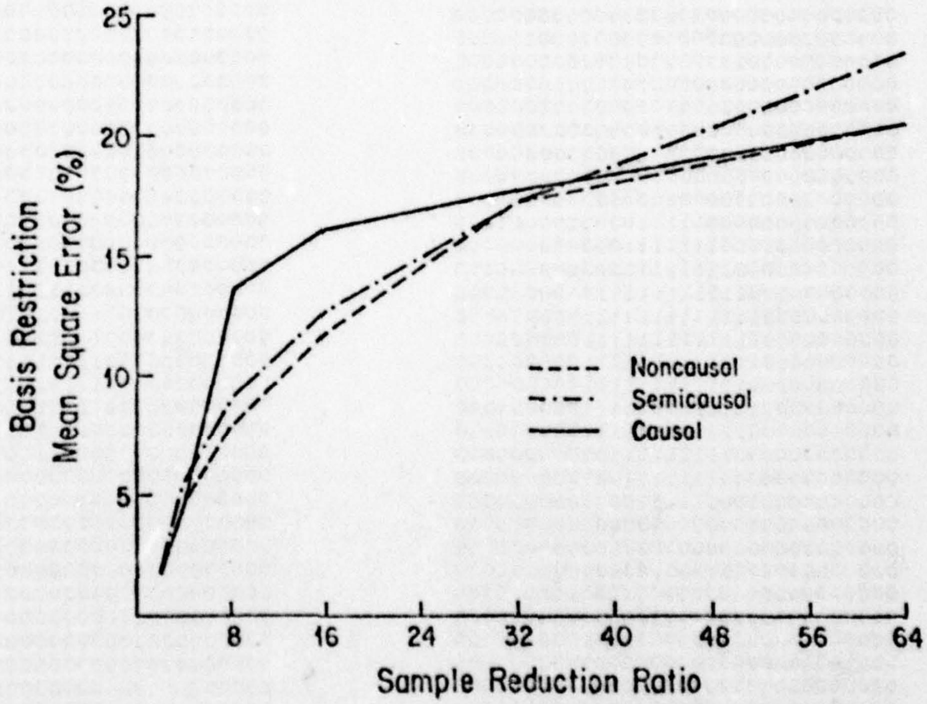


Figure 2.7 Basis Restriction Mean Square Error versus Sample Reduction Ratio for the Fourier Transform Model Zonal Sampling of a 256 × 256 Earth Image with 16 × 16 Image Block Size

(a) NC1, NC2-A, NC2-B,
8:1 Sample Reduction

(d) NC1, NC2-A, NC2-B,
16:1 Sample Reduction

(b) SC1,
8:1 Sample Reduction

(e) SC1,
16:1 Sample Reduction

(c) C1,
8:1 Sample Reduction

(f) C1,
16:1 Sample Reduction

Figure 2.8 Fourier Transform Model Zonal Sampled Girl Image with 16×16 Image Block Size

(a) NC1, NC2-A, NC2-B,
8:1 Sample Reduction

(b) SC1,
8:1 Sample Reduction

(c) C1,
8:1 Sample Reduction

(d) NC1, NC2-A, NC2-B,
16:1 Sample Reduction

(e) SC1,
16:1 Sample Reduction

(f) C1,
16:1 Sample Reduction

Figure 2.9 Fourier Transform Model Zonal Sampled Earth Image with

Table 2.1 Summary of Some PDE Models and Their Discrete Approximates $D[u] = \epsilon$ Used for Image Modeling

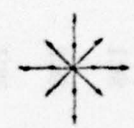
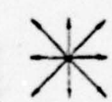
Model	PDE Operator	Discrete Operator $D(z_1,z_2)$	Spectral Density of ϵ $S_\epsilon(z_1,z_2)$	Spectral Density of u $S_u(z_1,z_2)$	Spatial Structure	Comments
Causal C1	$\frac{\partial^2}{\partial x \partial y} + a_1 \frac{\partial}{\partial x} + a_2 \frac{\partial}{\partial y} + a_3$	$(1-\rho z_1^{-1})(1-\rho z_2^{-1})$	$(1-\rho^2)^2 = \beta^2$	$\frac{(1-\rho^2)^2}{(1-\rho z_1)(1-\rho z_1^{-1})(1-\rho z_2)(1-\rho z_2^{-1})}$		Hyperbolic wave equation Covariance $R(k,l) = \sigma^2 \rho^{\lvert k \rvert + \lvert l \rvert}$
Semicausal SC1	$\frac{\partial}{\partial y} - \frac{\partial^2}{\partial x^2} + a_1$	$1-\alpha z_1 - \alpha z_1^{-1} - \gamma z_2^{-1}$	β^2	$\frac{\beta^2}{(1-\alpha z_1 - \alpha z_1^{-1} - \gamma z_2^{-1})(1-\alpha z_1^{-1} - \alpha z_1 - \gamma z_2)}$		Parabolic diffusion equation White noise driven model
Semicausal SC2	$(\frac{\partial^2}{\partial x^2} + a_1)(\frac{\partial}{\partial y} + a_2)$	$1-\alpha(1-\rho z_2^{-1})(z_1+z_1^{-1})-\rho z_2^{-1}$	$\beta^2[1-\alpha(z_1+z_1^{-1})]$	Same as C1		$\alpha = \frac{\rho}{1+\rho^2}$, $\beta^2 = \frac{(1-\rho^2)^2}{(1+\rho^2)}$ Minimum variance model
Noncausal NC1	$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + a_1$	$1-\alpha(z_1+z_1^{-1}+z_2+z_2^{-1})$	$\beta^2[1-\alpha_1(z_1+z_1^{-1}+z_2+z_2^{-1})]$	$\frac{\beta^2[1-\alpha_1(z_1+z_1^{-1}+z_2+z_2^{-1})]}{[1-\alpha(z_1+z_1^{-1}+z_2+z_2^{-1})]^2}$		Elliptic potential equation Minimum variance model when $\alpha_1 = \alpha$
Noncausal NC2	$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)^2 + a_1$	$[1-\alpha(z_1+z_1^{-1}+z_2+z_2^{-1})]^2$	$\beta^2[1-\alpha_1(z_1+z_1^{-1}+z_2+z_2^{-1})]^2$	$\frac{\beta^2[1-\alpha_1(z_1+z_1^{-1}+z_2+z_2^{-1})]^2}{[1-\alpha(z_1+z_1^{-1}+z_2+z_2^{-1})]^4}$		Elliptic biharmonic equation Minimum variance model when $\alpha_1 = \alpha$ NC2-A when $\alpha_1 = \alpha$ NC2-B when $\alpha_1 = 0$
Noncausal NC3	$\left(\frac{\partial^2}{\partial x^2} + a_1\right)\left(\frac{\partial^2}{\partial y^2} + a_2\right)$	$[1-\alpha(z_1+z_1^{-1})][1-\alpha(z_2+z_2^{-1})]$	$\beta^2[1-\alpha(z_1+z_1^{-1})][1-\alpha(z_2+z_2^{-1})]$	Same as C1		$\beta^2 = \frac{(1-\rho^2)^2}{(1+\rho^2)^2}$, $\alpha = \frac{\rho}{1+\rho^2}$ Minimum variance model

Table 2.2 Actual Parameters Measured for Various Models of the 256 × 256 Image to Match the 16 × 16 Covariance Matrix of the Image

Image	Model	Parameters	ℓ.s.e.%
	C1	$\rho = 0.962$	0.720
	SC1	$\alpha = 0.4275,\ \gamma = 0.1415,\ \beta^2 = 0.0198$	0.162
	SC2	$\rho = 0.962,\ \alpha = 0.49963,\ \beta^2 = 0.0029$	0.720
Girl	NC1	$\alpha = 0.2494,\ \alpha_1 = 0.95\alpha,\ \beta^2 = 0.0744$	0.377
	NC2-A	$\alpha = 0.2494,\ \alpha_1 = \alpha,\ \beta^2 = 0.0053$	0.598
	NC2-B	$\alpha = 0.2476,\ \alpha_1 = 0,\ \beta^2 = 8 \times 10^{-6}$	1.697
	NC3	$\rho = 0.962,\ \alpha = 0.49963,\ \beta^2 = 0.0015$	0.720
	C1	$\rho = 0.968$	0.891
	SC1	$\alpha = 0.4275,\ \gamma = 0.1420,\ \beta^2 = 0.0169$	0.572
	SC2	$\rho = 0.968,\ \gamma = 0.49974,\ \beta^2 = 0.0020$	0.891
Earth	NC1	$\alpha = 0.24966,\ \alpha_1 = 0.95\alpha,\ \beta^2 = 0.00631$	0.303
	NC2-A	$\alpha = 0.24955,\ \alpha_1 = \alpha,\ \beta^2 = 0.0073$	1.353
	NC2-B	$\alpha = 0.2482,\ \alpha_1 = 0,\ \beta^2 = 3 \times 10^{-6}$	2.877
	NC3	$\rho = 0.968,\ \alpha = 0.49974,\ \beta^2 = 9.7 \times 10^{-4}$	0.891
	C1	$\rho = 0.9536$	2.158
	SC1	$\alpha = 0.4275,\ \gamma = 0.1420,\ \beta^2 = 0.0169$	0.827
	SC2	$\rho = 0.9536,\ \alpha = 0.49944,\ \beta^2 = 0.0043$	2.158
Chemical Plant	NC1	$\alpha = 0.24966,\ \alpha_1 = 0.95\alpha,\ \beta^2 = 0.0631$	0.401
	NC2-A	$\alpha = 0.02493,\ \alpha_1 = \alpha,\ \beta^2 = 0.0084$	3.677
	NC2-B	$\alpha = 0.24708,\ \alpha_1 = 0,\ \beta^2 = 1.5 \times 10^{-3}$	6.724
	NC3	$\rho = 0.9536,\ \alpha = 0.49944,\ \beta^2 = 0.00225$	2.158

Table 2.3 Basis Restriction Mean Square Error of the Girl Image Represented by:

Sample Reduction Ratio	Size of an Image Block 8 x 8	16 x 16	32 x 32	64 x 64
(a) Causal Image Model (C1)				
2	0.774%	0.693%	0.588%	0.569%
4	3.611%	2.226%	2.030%	1.914%
8	5.540%	8.984%	4.604%	4.656%
16	7.734%	11.133%	13.662%	7.760%
32		13.208%	14.975%	11.740%
64		16.553%	22.961%	17.758%
128			25.953%	23.151%
256			31.154%	30.795%
512				49.018%
1024				60.951%
(b) Semicausal Image Model (SC1)				
2	1.334%	1.398%	1.311%	0.943%
4	3.264%	3.677%	3.104%	2.463%
8	6.056%	5.989%	5.182%	4.133%
16	10.931%	9.258%	8.121%	6.543%
32		13.791%	12.521%	10.007%
64		19.430%	18.253%	14.530%
128			26.089%	22.249%
256			39.802%	32.243%
512				39.690%
1024				60.951%
(c) Noncausal Image Model (NC1 or NC2-A or NC2-B)				
2	1.195%	0.887%	0.684%	0.571%
4	3.062%	3.066%	2.573%	1.977%
8	6.232%	5.291%	4.909%	3.785%
16	7.734%	8.551%	8.189%	6.474%
32		13.791%	12.132%	10.002%
64		16.553%	17.553%	14.745%
128			26.089%	21.272%
256			31.154%	30.640%
512				39.690%
1024				51.726%

Table 2.4 Basis Restriction Mean Square Error of the Earth Image Represented by:

Sample Reduction Ratio	8 x 8	16 x 16	32 x 32	64 x 64
	Size of an Image Block			
(a) Causal Image Model (C1)				
2	1.989%	1.660%	1.510%	1.476%
4	7.166%	4.571%	4.596%	4.450%
8	10.141%	13.554%	8.307%	8.881%
16	12.093%	16.132%	18.504%	13.064%
32		17.797%	19.399%	18.863%
64		20.534%	25.325%	22.981%
128			27.203%	27.440%
256			31.288%	33.120%
512				37.983%
1024				46.098%
(b) Semicausal Image Model (SC1)				
2	2.126%	1.896%	1.765%	1.546%
4	5.773%	5.441%	4.746%	4.513%
8	10.263%	9.049%	7.732%	7.337%
16	16.062%	12.679%	11.122%	10.751%
32		17.338%	15.614%	14.916%
64		23.428%	21.699%	20.342%
128			27.784%	26.163%
256			33.762%	32.118%
512				38.608%
1024				44.813%
(c) Noncausal Image Model (NC1 or NC2-A or NC2-B)				
2	2.247%	1.723%	1.490%	1.389%
4	5.330%	4.752%	4.186%	3.939%
8	9.659%	7.914%	7.175%	6.725%
16	12.093%	11.884%	10.913%	10.346%
32		17.338%	15.445%	14.693%
64		20.540%	20.402%	19.845%
128			27.784%	25.977%
256			31.288%	31.340%
512				38.608%
1024				41.930%

CHAPTER THREE

SEMICAUSAL MODELS AND HYBRID CODING

3.1 Introduction

In this chapter we consider image coding techniques which combine transform and predictive coding techniques. Such techniques have been called Hybrid coding [17] or, more generally, Hybrid processing. The basic idea behind hybrid processing is to first convert a two-dimensional discrete random field (an array of random variables) into a sequence of one-dimensional, independent, discrete random processes via a unitary transformation. Then each of these one-dimensional processes is processed independently by one-dimensional techniques such as recursive filtering (for image restoration applications) or predictive coding (such as DPCM, for image transmission/storage applications).

In this chapter we show the semicausal models considered earlier yield hybrid coding algorithms. Once the parameters of a semicausal model are identified, the entire hybrid coding scheme can be designed. Adaptive hybrid coding schemes can then be designed by simply adapting the semicausal model to changes in image statistics. Some practical adaptive hybrid coding methods which strike a balance between complexity and performance are studied. Application of these models to coding of noisy images are also considered.

3.2 Semicausal Image Models and Hybrid Coding

Let $\{u_{i,j}\}$ represent a zero mean, Caussian, stationary random field whose covariance function is defined as

$$r(k,\ell) = Eu_{i,j}u_{i+k,j+\ell} \quad \forall\ i,j. \qquad (3.2\text{-}1)$$

Consider an $N \times M$ segment of this random field where $1 \leq i \leq N$ and $1 \leq j \leq M$, and let u_j denote the jth column of the image, i.e.,

$$u_j = [u_{1,j} u_{2,j} \cdots u_{N,j}]^T \ . \qquad (3.2\text{-}2)$$

From (3.2-1) it follows that

$$E u_j u_{j+\ell} = R_\ell \qquad (3.2\text{-}3a)$$

where R_ℓ is an $N \times N$ Toeplitz matrix of elements

$$R_\ell(m,n) = r(m-n,\ell) \qquad 1 \leq m,n \leq N \ . \qquad (3.2\text{-}3b)$$

3.2.1 Hybrid Coding of Separable Covariance Image Fields

Consider the often used separable covariance model for images

$$r(k,\ell) = \sigma^2 \rho_v^{|k|} \rho_h^{|\ell|} \qquad (3.2.1\text{-}1)$$

where ρ_v and ρ_h are the vertical and horizontal correlations, respectively.

This gives

$$R_\ell = \sigma^2 \rho_h^{|\ell|} R_v \qquad (3.2.1\text{-}2a)$$

where

$$R_v = \{\rho_v^{|m-n|}\} \ . \qquad (3.2.1\text{-}2b)$$

It is well known that R_v is the covariance of a zero mean, first order stationary Markov process whose one step correlation is ρ_v. Let Ψ denote the Karhunen Loeve (KL) transform of this process (see Chapter 4), i.e.,

Ψ is a unitary matrix that satisfies

$$\Psi R_v \Psi^T = \Lambda \qquad \Psi\Psi^T = I \tag{3.2.1-3}$$

where $\Lambda = \{\lambda_i\}$ is the diagonal matrix of eigenvalues of R_v. Define the transformation

$$v_j \triangleq \Psi u_j . \tag{3.2.1-4}$$

Using (3.2.1-2a), (3.2.1-3) and (3.2.1-4), one obtains

$$Ev_j v_{j+\ell}^T = \Psi R_\ell \Psi^T = \sigma^2 \rho_h^{|\ell|} \Lambda \tag{3.2.1-5a}$$

Or $$Ev_j(i) v_{j+\ell}(i+k) = \sigma^2 \rho_h^{|\ell|} \lambda_i \delta_{k,0} . \tag{3.2.1-5b}$$

Eqn. (3.2.1-5b) implies that i) the elements of the transformed column v_j are mutually uncorrelated and ii) for each i, the sequence $\{v_j(i),\ 1 \le j \le M\}$ is a first order, stationary Markov process. Therefore, we can write

$$v_j(i) = \rho_h v_{j-1}(i) + e_j(i) \tag{3.2.1-6a}$$

where

$$Ee_j(i) = 0 \qquad Ee_j(i) e_{j+\ell}(i+k) = \sigma^2(1-\rho_h^2)\lambda_i \delta_{\ell,0} \delta_{k,0} \tag{3.2.1-6b}$$

and $\{v_j(i),\ 1 \le i \le N\}$ are the elements of the vector v_j. (3.2.1-6) spells the so called Hybrid coding method for the random field $\{u_{i,j}\}$. First, each column u_j is unitarily transformed (ideally by the KL transform) to obtain v_j and for each sequence $\{v_j(i),\ j = 1,2,\ldots\}$ for $1 \le i \le N$, an independent DPCM channel is used. This overall scheme is shown in Fig. 3.1. Fig. 3.2 shows the details of the ith DPCM channel and the

reconstruction filter at the receiver. Often, a fast unitary transform is used to replace the KL transform to make the scheme more practical (see Chapter 4). For video image data where $\rho_h \approx \rho_v \approx 0.95$, the Cosine transform performs very close to the KL transform defined in (3.2.1-3) [1,31,34]

More generally, if any two-dimensional random field $\{u_{i,j}\}$ can be transformed via a unitary transformation to a sequence of independent one-dimensional Markov processes, then a Hybrid coding scheme for transmissions of $\{u_{i,j}\}$ can be designed.

3.2.2 Semicausal Representations as Two Source Models for Finite Random Fields

Consider the given image to be an $N \times M$ segment of the infinite, stationary random field. Semicausal representations we have considered in the previous chapter [see (2.5.2-1) and 2.5.2-9)] when written in an $N \times 1$ vector notation are of the form

$$Qu_j = Pu_{j-1} + \varepsilon_j + b_j \tag{3.2.2-1}$$

where Q is a tridiagonal Toeplitz matrix defined by

$$q_{i,j} = \begin{cases} 1, & i = j \\ -\alpha, & |i-j| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.2-2a}$$

and

$$P = \begin{cases} \gamma I, & \text{for SC1} \\ \rho_h Q, & \text{for SC2} \end{cases} \tag{3.2.2-2b}$$

$$b_j = \begin{cases} \alpha[u_{0,j}, \overbrace{0, \ldots, 0}^{|N-2 \text{ terms}|}, u_{N+1,j}]^T, & \text{for SC1} \\ \alpha[b_{1,j}, \underbrace{0, \ldots, 0}_{|N-2 \text{ terms}|}, b_{N,j}]^T, & \text{for SC2} \end{cases} \tag{3.2.2-2c}$$

$$b_{1,j} = u_{0,j} - \rho_h u_{0,j-1}.$$

$$b_{N,j} = u_{N+1,j} - \rho_h u_{N+1,j-1}. \tag{3.2.2-2d}$$

The $N \times 1$ vector b_j contains only two nonzero entries which depend only on the boundary variables $\{u_{0,j}\}$ and $\{u_{N+1,j}\}$ of the jth image column. *Note from (3.2-2) that the jth image column, u_j contains only N elements.* The covariance of ε_j is

$$R_\varepsilon = E\varepsilon_j\varepsilon_k^T = \begin{cases} \beta^2 I\delta_{j,k}, & \text{for SC1} \\ \beta^2 Q\delta_{j,k}, & \text{for SC2} \end{cases} \tag{3.2.2-2e}$$

where $\beta^2 = (1 - \rho_v^2)(1 - \rho_h^2) \big/ (1 + \rho_v^2)$ for the SC2 model. (3.2.2-2f)

Eqn. (3.2.2-1) can be looked at, in general, as a two source model for the vector sequence $\{u_j\}$, whose statistics are specified by the statistics for the "source inputs", $\{\varepsilon_j\}$ and $\{b_j\}$. We note that (3.2.2-1) may not always qualify to be a vector Markov model

$$Qu_j = Pu_{j-1} = f_j \tag{3.2.2-3a}$$

by defining

$$f_j = \varepsilon_j + b_j. \tag{3.2.2-3b}$$

This is because the sequence $\{f_j\}$ is not guaranteed to be white, even though $\{\varepsilon_j\}$ is such. From the linearity of (3.2.2-1), the two source outputs can be identified by the solution of the equations

$$Qu_j^o = Pu_{j-1}^o + \varepsilon_j \tag{3.2.2-4a}$$

$$Qu_j^b = Pu_{j-1}^b + b_j \tag{3.2.2-4b}$$

and the image field is the composition [31]

$$u_j = u_j^o + u_j^b . \tag{3.2.2-5}$$

The second source output u_j^b depends completely on the boundary variables. Hence, as the size of the image column gets larger, one would expect a smaller contribution of u_j^b in influencing the statistical properties of u_j. *The* $N \times M$ *random field will be stationary when the statistics of the* boundary variables are consistent with those of $\{u_{i,j}\}$. In other words, the case of stationary statistics is only a special case of (3.2.2-5). For minimum variance representations, the orthogonality condition (2.3.1-1) implies that the two source outputs u_j^o and u_j^b are orthogonal. For stationary, white noise driven, semicausal models, one does not have this condition. If one starts by specifying the boundary variables to be orthogonal to the stationary white noise field $\{\varepsilon_{i,j}\}$, then the finite $N \times M$ field $\{u_{i,j}\}$ need not be stationary. Asymptotically, $(N,M \to \infty)$, (3.2.2-4) and (3.2.2-5) will yield stationary random field $\{u_{i,j}\}$ (assuming Q is always nonsingular and (3.2.2-1) is a stable recursive relation).

3.2.3 Stochastic Decoupling of Semicausal Models by the Cosine Transform

For a large number of image fields, the adjacent elements of a column (or a row) are highly correlated and often the correlation parameter has a value around 0.95. Thus, for a finite $N \times M$ random field, it is reasonable to assume that the boundary variables that need to be specified in (3.2.2-1) satisfy the conditions

$$u_{0,j} = u_{1,j}$$

$$u_{N+1,j} = u_{N,j} \quad . \qquad (3.2.3\text{-}1)$$

This means the two outermost boundary rows of the $(N+2) \times M$ field $\{u_{i,j};\ 0 \leq i \leq N+1,\ 1 \leq j \leq M\}$ are equal. This also means that (3.2.2-1) subject to (3.2.3-1) is no longer a stationary field on the $(N+2) \times M$ array of points. Using (3.2.3-1) in (3.2.3-1) and rearranging terms, we get a one source model (the second source is deterministic, as defined by (3.2.3-1))

$$Q_c u_j = P u_{j-1} + \varepsilon_j \qquad (3.2.3\text{-}2)$$

where Q_c is defined as

$$Q_c \triangleq \begin{bmatrix} 1-\alpha & -\alpha & & \bigcirc \\ -\alpha & 1 & \ddots & \\ & \ddots & \ddots & -\alpha \\ & & 1 & \\ \bigcirc & & -\alpha & 1-\alpha \end{bmatrix} \qquad (3.2.3\text{-}3a)$$

and

$$P = \begin{cases} \gamma I, & \text{for SC1} \\ \rho_h Q_c, & \text{for SC2}. \end{cases} \tag{3.2.3-3b}$$

For SC1 model, $\{\varepsilon_{i,j}\}$ is still assumed to be a stationary white noise field. For SC2 model, (3.2.3-2) should satisfy the minimum variance conditions of (2.3.1-1). When applied to (3.2.3-2), this implies

$$E\varepsilon_j \varepsilon_k^T = \beta^2 Q_c \delta_{j,k}.$$

Hence, the covariance of ε_j is now given as

$$R_\varepsilon = E\varepsilon_j \varepsilon_k^T = \begin{cases} \beta^2 I \delta_{j,k}, & \text{for SC1} \\ \beta^2 Q_c \sigma_{j,k}, & \text{for SC2}. \end{cases} \tag{3.2.3-4}$$

<u>Proposition</u>

The KL transform of any vector u_j of the random field of (3.2.3-2) is the discrete Cosine transform (DCT) [1] for SC1 as well as SC2 models. For proof, see [35].

The eigenvalues of Q_c are given by [see (4.2-11c)]

$$\lambda_{ci} = 1 - 2\alpha \cos\frac{(i-1)\pi}{N} \qquad 1 \le i \le N. \tag{3.2.3-5}$$

Define Ψ_c to be the DCT matrix. Since it diagonalizes the matrices Q_c and P we have

$$\Psi_c Q_c \Psi_c^T = \Lambda_c = \text{Diagonal } \{\lambda_{ci}\} \tag{3.2.3-6a}$$

$$\Psi_c P \Psi_c^T = \Gamma_c = \{\gamma_i\} \qquad \gamma_i = \begin{cases} \gamma & , \text{ for SC1} \\ \rho_h \lambda_{ci} & , \text{ for SC2} \end{cases} \tag{3.2.3-6b}$$

Also, define

$$v_j = \Psi_c u_j . \tag{3.2.3-7}$$

Multiplying both sides of (3.2.3-2) by Ψ_c and using (3.2.3-6) and (3.2.3-7) we obtain

$$\Lambda_c v_j = \Gamma v_{j-1} + e_j \tag{3.2.3-8a}$$

where

$$e_j \triangleq \Psi_c \varepsilon_j . \tag{3.2.3-8b}$$

Since Λ_c and Γ are diagonal, (3.2.3-8a) is a set of decoupled equations

$$\lambda_{ci} v_j(i) = \gamma_i v_{j-1}(i) + e_j(i) \qquad 1 \le i \le N. \tag{3.2.3-9}$$

From (3.2.3-4) and (3.2.3-6a) and (3.2.3-8b), $\{e_j(i)\}$ is a white noise field which satisfies the equation

$$Ee_j(i)e_k(\ell) = \begin{cases} \beta^2 \delta_{j,k} \delta_{i,\ell} & , \text{ for SC1} \\ \beta^2 \lambda_{ci} \delta_{j,k} \delta_{i,\ell} & , \text{ for SC2} \end{cases} . \tag{3.2.3-10}$$

Hence, for each i, the sequence $\{v_j(i), \forall j\}$ is a first order Markov sequence. Moreover, if the initial variables $\{v_0(i), 1 \le i \le N\}$ are uncorrelated then for $\forall j > 0$, $\{v_j(i), 1 \le i \le N\}$ will be uncorrelated elements of the vector v_j. Also, for $u_0 = 0$ or $u_0 = u_\infty$ (i.e., steady state in j), the above condition will always be satisfied. For arbitrary

initial conditions, say $u_0 = c$, where c is an arbitrary random vector, u_j will have a decomposition

$$u_j = u_j^o + u_j^c \tag{3.2.3-11a}$$

where $$Q_c u_j^o = P u_{j-1}^o + \varepsilon_j \qquad u_0^o = 0 \tag{3.2.3-11b}$$

and $$Q_c u_j^c = P u_{j-1}^c \qquad u_0 = c\ . \tag{3.2.3-11c}$$

The KL transform of u_j^o now will be the DCT. In a practical DPCM system, the statistics of the initial condition do not affect the ultimate system performance because the initial vector can always be quantized accurately and transmitted before starting the DPCM transmission.

3.2.4 Stochastic Decoupling of Stationary Semicausal Fields

Random fields for which (3.2.3-1) is not a reasonable assumption (i.e., one step correlation in the "j" direction is not close to 1) or when stationarity of the random field is an essential requirement of the model, we have to restrict ourselves to (3.2.2-1). The boundary variables (elements of b_j) are random variables which are samples from the stationary random field $\{u_{i,j}\}$ and cannot be specified independently. In that case, (3.2.2-1) admits the decomposition [see (3.2.2-4) - (3.2.2-5)]

$$Q u_j^o = P u_{j-1}^o + \varepsilon_j \qquad u_0^o = 0 \qquad j \geq 1 \tag{3.2.4-1a}$$

$$Q u_j^b = P u_{j-1}^b + b_j \qquad u_0^b = u_0 \qquad j \geq 1 \tag{3.2.4-1b}$$

where

$$u_j = u_j^o + u_j^b \qquad j \geq 1 \tag{3.2.4-2}$$

and u_0 is the initial value of the sequence $\{u_j\}$.

Proposition

The KL transform of any column vector u_j^o, $j \geq 1$ of the random field $\{u_{i,j}^o\}$ of (3.2.4-1a) is the Sine transform for SC1 as well as the SC2 models. For proof see [35].

Let Ψ_s be the Sine transform. Then

$$\Psi_s Q \Psi_s = \Lambda_s = \text{Diagonal } \{\lambda_i\} \quad \lambda_i = 1-2\alpha\cos\frac{i\pi}{N+1} \quad 1 \leq i \leq N \tag{3.2.4-3}$$

$$\Psi_s P \Psi_s = \Gamma = \{\gamma_i\} \qquad \gamma_i = \begin{cases} \gamma , & \text{for SC1} \\ \rho_h \lambda_i , & \text{for SC2} \end{cases} \tag{3.2.4-4}$$

Defining

$$v_j = \Psi_s u_j^o \qquad e_j = \Psi_s \varepsilon_j \tag{3.2.4-5}$$

and multiplying both sides of (3.2.4-1a) by Ψ_s and using (3.2.4-3), (3.2.4-4) and (3.2.4-5) we get a sequence of decoupled equations

$$\lambda_i v_j(i) = \gamma_i v_{j-1}(i) + e_j(i) \qquad v_0(i) = 0 \qquad 1 \leq i \leq N,\ j \geq 1 \tag{3.2.4-6a}$$

$$Ee_j(i)e_k(\ell) = \begin{cases} \beta^2 \delta_{i,\ell}\delta_{j,k} , & \text{for SC1} \\ \beta^2 \lambda_i \delta_{i,\ell}\delta_{j,k}, & \text{for SC2 .} \end{cases} \tag{3.2.4-6b}$$

From (3.2.4-6a) and (3.2.4-6b) it is evident that $\{v_j(i)\}$ is a sequence of N, first order Markov processes and for any $j \geq 1$ the elements $\{v_j(i), 1 \leq i \leq N\}$ are uncorrelated. Hence, Ψ_s is the KL transform of u_j^o. Figure 3.3 shows the realization of decomposition of (3.2.4-2) and stochastic decoupling of (3.2.4-6).

3.3 Hybrid Coder Design for Semicausal Random Fields

For the sake of simplicity of presentation, we will only consider the semicausal models of (3.2.3-2) which are decoupled by the Cosine transform. The sequence of decoupled equations (3.2.3-9) and (3.2.3-10) can be rewritten as

$$v_j(i) = \rho_i v_{j-1}(i) + e_j(i) \qquad 1 \le i \le N \tag{3.3-1}$$

where

$$\rho_i = \frac{\gamma_i}{\lambda_{ci}} = \begin{cases} \gamma/\lambda_{ci} \;, & \text{for SC1} \\ \rho_h \;, & \text{for SC2} \end{cases} \tag{3.3-2a}$$

and $\{e_j(i)\}$ is a white noise field whose covariances are given by

$$Ee_j(i)e_k(\ell) \triangleq \beta^2(i)\delta_{j,k}\delta_{i,\ell}$$

$$\beta^2(i) \triangleq \begin{cases} \beta^2/\lambda_{ci}^2 \;, & \text{for SC1} \\ \beta^2/\lambda_{ci} \;, & \text{for SC2} \;. \end{cases} \tag{3.3-2b}$$

3.3.1 DPCM Equations

Since (3.3-1) is a sequence of N independent Markov processes, each of these sequences could be transmitted via an independent DPCM channel, as shown in Fig. 3.2. The equation at the transmitter and receiver are simply as follows:

<u>Predictor:</u> $$\bar{v}_j(i) = \rho_i v^*_{j-1}(i) \tag{3.3.1-1a}$$

<u>Differential Signal:</u> $$\hat{e}_j(i) = v_j(i) - \bar{v}_j(i) \tag{3.3.1-1b}$$

Quantizer Output: $e_j^*(i)$ (3.3.1-1c)

Reconstruction Filter: $v_j^*(i) = \rho_i v_{j-1}^*(i) + e_j^*(i)$ (3.3.1-1d)

The encoding scheme requires, first, to take the Cosine transform of each column vector u_j. The DCT is a fast transform and is also the KL transform of each column of the semicausal image fields characterized by (3.2.3-2). This is followed by N DPCM channels for predictive coding of successive Cosine transformed vectors. The receiver simply reconstructs the transformed vectors according to (3.3.1-1d) and performs the inverse Cosine transformation. To complete the design, we now need to specify the quantizers in the various DPCM channels.

3.3.2 Bit Allocation

We will assume all the quantizers are *identical in their* characteristics (i.e., bit rate vs. distortion). Let

P = Average desired bit rate in bits/pixel of the image field.

$f(n)$ = Mean square distortion of a quantizer with 2^n levels for unit variance input random variables. This is a positive, monotonically decreasing convex function. We will assume $f(n)$ has a continuous first derivative.

$0 \le n_i$ = Number of bits per pixel, allocated to the ith DPCM channel.

From the definition of P, we have

$$\frac{1}{N}\sum_{i=1}^{N} n_i = P \qquad n_i \ge 0 . \tag{3.3.2-1}$$

The average mean square distortion in encoding of the image vector u_j is given by

$$D = \frac{1}{N} E\delta u_j^T \delta u_j \qquad (3.3.2\text{-}2)$$

where $\delta u_j \overset{\Delta}{=} u_j - u_j^*$.

Since $v_j = \Psi_c u_j$ is a unitary transformation and $v_j(i)$ are coded via DPCM, we have

$$\begin{aligned}\delta v_j(i) &= v_j(i) - v_j^*(i)\\ &= v_j(i) - \rho_i v_{j-1}^*(i) - e_j^*(i)\\ &= \hat{e}_j(i) - e_j^*(i)\\ &= \delta\hat{e}_j(i) \qquad (3.3.2\text{-}3)\end{aligned}$$

i.e., the reconstruction error in $v_j(i)$ (assuming noiseless channel) equals the quantization error in the ith channel. This gives

$$\begin{aligned}D &= \frac{1}{N} E\delta v_j^T \delta v_j\\ &= \frac{1}{N} \sum_{i=1}^{N} E(\delta e_j(i))^2 . \qquad (3.3.2\text{-}4)\end{aligned}$$

Thus, the average distortion per pixel is the average quantizer distortion. If $\hat{\beta}_j^2(i)$ denotes the variance of the jth quantizer input in the ith DPCM channel, then

$$E(\delta\hat{e}_j(i))^2 = \hat{\beta}_j^2(i) f(n_i) \quad \forall\, j . \qquad (3.3.2\text{-}5)$$

Using (3.3-1), (3.3.1-1a) *and* (3.3.2-4) *in* (3.3.1-1b) *and simplifying,* we get

$$\hat{e}_j(i) = v_j(i) - \bar{v}_j(i)$$
$$= e_j(i) + \rho_i \delta\hat{e}_{j-1}(i) \ . \qquad (3.3.2\text{-}6)$$

Eqns. (3.3.2-5) and (3.3.2-6) together give a recursive relation for the variance of $\hat{e}_j(i)$ as

$$\hat{\beta}_j^2(i) = \beta^2(i) + \rho_i^2 \hat{\beta}_{j-1}^2(i) f(n_i) \qquad \hat{\beta}_0^2 = 0 \qquad (3.3.2\text{-}7)$$

where we have used the fact that $e_j(i)$ is uncorrelated with the quantization error at step j-1. In steady state

$$\hat{\beta}_j^2(i) \triangleq \hat{\beta}^2(i) = \frac{\beta^2(i)}{1 - \rho_i^2 f(n_i)} \ . \qquad (3.3.2\text{-}8)$$

Hence, (3.3.2-5) becomes

$$E(\delta\hat{e}_j(i))^2 = \frac{\beta^2(i) f(n_i)}{1 - \rho_i^2 f(n_i)} \qquad (3.3.2\text{-}9)$$

giving

$$D = \frac{1}{N} \sum_{i=1}^{N} g_i(n_i) \beta^2(i) \qquad (3.3.2\text{-}10a)$$

where

$$g_i(x) = \frac{f(x)}{1 - \rho_i^2 f(x)} \qquad x \geq 0 \ . \qquad (3.3.2\text{-}10b)$$

The function $g_i(x)$ represents the mean square distortion of the ith DPCM loop of a Markov process whose prediction error variance is unity. Like the function f(x), for $|\rho_i| < 1$, each $g_i(x)$ is also a positive, monotonically decreasing, convex function with a continuous first derivative.

The bit allocation problem is to minimize (3.3.2-10a) subject to the constraints of (3.3.2-1). This is equivalent to finding

$$D' = \min_{\substack{n_i \geq 0 \\ \lambda}} \left\{ \lambda P - \frac{1}{N} \sum_{i=1}^{N} [g_i(n_i)\beta^2(i) + \lambda n_i] \right\}$$

$$= \min_{\lambda} \left\{ \lambda P - \frac{1}{N} \sum_{i=1}^{N} \max_{n_i \geq 0} [g_i(n_i)\beta^2(i) + \lambda n_i] \right\} . \qquad (3.3.2\text{-}11)$$

Solution of constrained minimization problems of this type are well known in optimization theory. Recently, Segall [66] has considered a similar bit allocation problem [with $g_i(x) = g(x)$, $\forall$ i] for encoding of vector sources. The solution is given by

$$g_i'(n_i)\beta^2(i) + \lambda = 0 \qquad \lambda \leq -g_i'(0)\beta^2(i)$$

$$n_i = 0 \qquad \lambda > -g_i'(0)\beta^2(i) . \qquad (3.3.2\text{-}12)$$

Denoting $$h_i(x) = g_i'^{-1}(x) \qquad (3.3.2\text{-}13)$$

we can solve (3.3.2-12) for n_i as

$$n_i = \begin{cases} h_i\left(\dfrac{-\lambda}{\beta^2(i)}\right) , & \lambda \leq -g_i'(0)\beta^2(i) \\ 0 , & \lambda > -g_i'(0)\beta^2(i) \end{cases} \qquad (3.3.2\text{-}14)$$

where λ is the root of the nonlinear equation

$$\sum_{i:\lambda \leq -g_i'(0)\beta^2(i)} h\left(-\frac{\lambda}{\beta^2(i)}\right) = NP . \qquad (3.3.2\text{-}15)$$

The value of the minimum distortion is

$$D_{min} = \frac{1}{N}\left[\sum_{\lambda \leq -g_i'(0)\beta^2(i)} g_i(n_i)\beta^2(i) + \sum_{\lambda > -g_i'(0)\beta^2(i)} \beta^2(i)\right]. \quad (3.3.2\text{-}16)$$

Now consider the following special cases.

1. $\hat{\beta}^2(i) \approx \beta^2(i)$, i.e., the effect of quantization on the variance of the prediction error is ignored. This gives

$$g_i(n) \approx f(n) \quad \forall\, i\,. \quad (3.3.2\text{-}17)$$

When the quantizer in the DPCM loop is the ideal optimal block encoder (Shannon quantizer) that achieves the rate distortion bound we have

$$f(n) = 2^{-2n} \quad (3.3.2\text{-}18)$$

which gives

$$f'(x) = (-2\log_e 2)2^{-2x}$$

or $$h(x) = f'^{-1}(x) = -\tfrac{1}{2}\log_2\left(\frac{x}{-2\log_e 2}\right)\,. \quad (3.3.2\text{-}19)$$

Defining

$$\theta = \lambda/(2\log_e 2) \quad (3.3.2\text{-}20)$$

and using (3.3.2-18) in (3.3.2-14) to (3.3.2-16), we get

$$n_i = \begin{cases} \tfrac{1}{2}\log_2\left(\dfrac{\beta^2(i)}{\theta}\right), & 0 < \theta \leq \beta^2(i) \\ 0, & \theta > \beta^2(i) \end{cases} \quad (3.3.2\text{-}21a)$$

where θ is the solution of

$$\sum_{i=1}^{N} \max\left[0, \tfrac{1}{2}\log_2 \frac{\beta^2(i)}{\theta}\right] = NP \qquad (3.3.2\text{-}21b)$$

and the minimized distortion is

$$D_{min} = \frac{1}{N} \sum_{i=0}^{N} \min[\beta^2(i), \theta] \ . \qquad (3.3.2\text{-}21c)$$

Eqns. (3.3.2-21) are valid for both SC1 and SC2 models, where $\beta^2(i)$ is defined by (3.3-2b).

2. Consider the case when $f(\cdot)$ and $g_i(\cdot)$ are given by (3.3.2-18) and (3.3.2-10b) respectively. Then

$$g_i'(x) = \frac{f'(x)}{(1 - \rho_i^2 f(x))^2}$$

$$= \frac{(-2\log_e 2) 2^{-2x}}{(1 - \rho_i^2 2^{-2x})^2} \ . \qquad (3.3.2\text{-}22)$$

From this equation one can obtain a unique inverse (note that $g_i'(x)$ must be nonpositive for all $x \geq 0$), as (see Appendix C)

$$g_i'^{-1}(x) \triangleq h_i(x) = \begin{cases} -\tfrac{1}{2}\log_2 \left[\dfrac{-\log_e 2 + \rho_i^2 x + \sqrt{(\log_e 2)^2 - (2\log_e 2)\rho_i^2 x}}{\rho_i^4 x}\right], & \rho_i \neq 0 \\ -\tfrac{1}{2}\log_2 \left(\dfrac{-x}{2\log_e 2}\right), & \rho_i = 0 \ . \end{cases} \qquad (3.3.2\text{-}23)$$

For SC1 model, defining

$$\theta = \frac{\lambda}{(2\log_e 2)} \qquad (3.3.2\text{-}24)$$

and using (3.3.2-23) in (3.3.2-14), (3.3.2-15) and (3.3.2-16) we get

$$n_i = \begin{cases} \frac{1}{2}\log_2 \dfrac{\beta^2(i)}{\theta}\, w(i,\theta) & , \quad 0 < \theta \le \dfrac{\beta^2(i)}{(1-\rho_i^2)^2} \\ 0 & , \quad \theta > \dfrac{\beta^2(i)}{(1-\rho_i^2)^2} \end{cases} \tag{3.3.2-25a}$$

where

$$w(i,\theta) \triangleq \begin{cases} 1 \quad , \quad \rho_i = 0 \\ \dfrac{1}{2\rho_i^4}\left[1 + \dfrac{2\rho_i^2\theta}{\beta^2(i)} - \left(1 + \dfrac{4\rho_i^2\theta}{\beta^2(i)}\right)^{\frac{1}{2}}\right], \quad \rho_i \neq 0 \end{cases} \tag{3.3.2-25b}$$

θ is the solution of

$$\sum_{i:0<\theta\le \frac{\beta^2(i)}{(1-\rho_i^2)^2}} \frac{1}{2}\log_2 \frac{\beta^2(i)w(i,.)}{\theta} = NP \tag{3.3.2-25c}$$

and the minimum distortion is

$$D_{min} = \frac{1}{N}\sum_{i=1}^{N} \min[\beta^2(i),\ \theta/w(i,\theta)] \ . \tag{3.3.2-25d}$$

In the case of SC2 model, substitution of $\rho_i = \rho_h$ = constant, (3.3.2-25) above gives the desired bit allocation and distortion formulas. The bit allocations are obtained by first solving (3.3.2-25c) iteratively (by Newton's method, for example) to obtain θ, for a fixed rate P. Given θ, (3.3.2-25a) and (3.3.2-25b) yield the desired allocations $\{n_i\}$ and (3.3.2-25d) gives the corresponding distortion.

3.3.3 Integer Bit Allocation

In the above, we assumed n_i to be a real number. In many practical situations n_i is required to be an integer. An approximate integer bit allocation can be obtained via the foregoing method by rounding n_i to the nearest integer. The actual bit rate then would be

$$P_a = \frac{1}{N} \sum_{i=1}^{N} [n_i]$$

where [x] indicates the nearest integer to x. To obtain the exact solution, the minimization of (3.3.2-10a) is subject to the constraint that $\{n_i\}$ be a set of nonnegative integers. The solution is obtained via an iterative algorithm due to Fox [11] as follows.

Let $\underline{n}$ denote a vector of elements $\{n_i, \ 1 \le i \le N\}$.

1. Start with the allocation $\underline{n}^0 = \underline{0}$.
2. Set $\ell = 1$.
3. $\underline{n}^\ell \leftarrow \underline{n}^\ell + e_i$ where e_i is the ith unit vector and is any index for which

   $$\Delta_{i,\ell} = \beta^2(i) g_i(n_i^{\ell-1}) - \beta^2(i) g_i(n_i^{\ell-1} + 1)$$

   is maximum. The quantity $\Delta_{i,\ell}$ is called the marginal return of increasing the ith allocation in ℓth step by one bit.
4. If $\sum_{i=1}^{N} n_i^\ell \ge PN$, terminate; otherwise $\ell \leftarrow \ell + 1$ and go to step 3.

This algorithm simply means the marginal returns $\Delta_{i,\ell}$ are calculated for $1 \le i \le N$ and $\ell = 1,2,\ldots,$ and are arranged in a decreasing order until all the bits are exhausted. Tables 3.1 and 3.2 show the bit

allocations for the SC1, SC2 and separable covariance models for Girl image obtained via real bit allocation and integer bit allocation algorithms described above.

3.4 Adaptive Hybrid Coding

The coding scheme of the previous section can be adapted to image fields whose spatial statistics vary slowly by updating the various coefficients of the image model with variations of the statistics. A complete update of the image model parameters could often increase the complexity of the coder to make it impractical. In the sequel we consider two adaptive schemes which offer reasonable improvement in coder performance with only a marginal increase in its complexity.

3.4.1 Adaptive Variance Estimation

For a fixed predictor in the feedback loop of a DPCM channel, the variance of the prediction error will fluctuate with changes in image statistics. In this scheme we update the variance of the prediction error at each step j. This updated variance is used to adjust the spacing of the quantizer levels in each DPCM channel. For mean square error criterion, the decision and reconstruction levels of a quantizer are directly proportional to the standard deviation of the input random variables. Hence, the above mentioned adaptation can be achieved by simply normalizing the prediction error by its updated standard deviation. The quantizer levels are then fixed and correspond to unit variance random variables. Fig. 3.4 shows this scheme.

Let

$$\hat{\sigma}_j^2(i) \triangleq \text{Variance of } e_j(i)\text{, the prediction error at step j of the ith DPCM loop} = E[e_j(i)]^2 \qquad (3.4.1\text{-}1a)$$

$$\overset{*}{\sigma}{}_j^2(i) \triangleq \text{Variance of the quantized values} = E[e_j^*(i)]^2 \ . \qquad (3.4.1\text{-}1b)$$

Since the quantized variables $e_j^*(i)$ are available both at the receiver and the transmitter, it is easy to estimate $\overset{*}{\sigma}{}_j^2(i)$. Castellino et al [8] have suggested a formula for an estimate of $\overset{*}{\sigma}{}_j^2(i)$ (for single channel DPCM system) as

$$\overset{*}{\sigma}{}_j^2(i) = \frac{1-\gamma}{\gamma} \sum_{m=1}^{j} \gamma^m e_{j-m+1}^{*2}(i) + \gamma^j \overset{*}{\sigma}{}_1^2(i) \qquad 0 < \gamma < 1. \qquad (3.4.1\text{-}2)$$

This has been called an exponential average variance estimator. A more convenient form of (3.4.1-2) is the recursion

$$\overset{*}{\sigma}{}_{j+1}^2(i) = (1-\gamma)e_j^{*2}(i) + \gamma\overset{*}{\sigma}{}_j^2(i) \qquad j = 1,2,\ldots \qquad (3.4.1\text{-}3)$$

For small quantization errors one may take $\overset{*}{\sigma}_j$ to be equal to $\hat{\sigma}_j$. For Lloyd-Max quantizers [44] a more accurate estimate of $\hat{\sigma}_j$ is possible. Since the variance of a Lloyd-Max quantizer input equals the sum of the variances of the quantizer output and the quantization error, we have

$$\hat{\sigma}_j^2(i) = \overset{*}{\sigma}{}_j^2(i) + f(n_i)\hat{\sigma}_j^2(i)$$

or

$$\hat{\sigma}_j^2(i) = \frac{\overset{*}{\sigma}{}_j^2(i)}{1-f(n_i)} \qquad n_i \geq 1 \qquad (3.4.1\text{-}4)$$

where n_i is the number of bits allocated to the ith channel. Using (3.4.1-4) in (3.4.1-3) we get

$$\hat{\sigma}^2_{j+1}(i) = \frac{1-\gamma}{1-f(n_i)} \; e^{*2}_j(i) + \gamma\hat{\sigma}^2_j(i) \qquad n_i > 1 \quad 0 < \gamma < 1. \quad (3.4.1\text{-}5)$$

The above estimate become poor for DPCM channels which are assigned small number of bits ($n_i \approx 1$). For these channels $\hat{\sigma}_j(i)$ could be estimated by some sort of extrapolation of the already estimated $\hat{\sigma}_j(i)$. For example, for the SC1 model, we know

$$\hat{\sigma}^2_j(i) \approx E\,\hat{e}^2_j(i) = \frac{\beta^2}{[1-\rho^2_i f(n_i)]\lambda^2_{ci}} \qquad i \in I \qquad (3.4.1\text{-}6)$$

where I is the set of, say, m channels for which $n_i > 1$. From (3.4.1-6) an estimate of β^2 at step j, denoted by $\tilde{\beta}^2_j$ is given by

$$\tilde{\beta}^2_j = \frac{1}{m} \sum_{i\varepsilon I} \hat{\sigma}^2_j(i)\lambda^2_{ci}[1 - \rho^2_i f(n_i)] \; . \qquad (3.4.1\text{-}7)$$

This gives the extrapolated estimates as

$$\hat{\sigma}^2_j(i) = \frac{\tilde{\beta}^2_j}{\lambda^2_{ci}[1-\rho^2_i f(n_i)]} \qquad i \notin I \; . \qquad (3.4.1\text{-}8)$$

For SC2 model, the equations corresponding to (3.4.1-7) and (3.4.1-8) become

$$\tilde{\beta}^2_j = \frac{1}{m} \sum_{i\varepsilon I} \hat{\sigma}^2_j(i)\lambda_{ci}[1-\rho^2_h f(n_i)] \qquad (3.4.1\text{-}9)$$

$$\hat{\sigma}^2_j(i) = \frac{\tilde{\beta}^2_j}{\lambda_{ci}[1-\rho^2_h f(n_i)]} \qquad i \notin I \; . \qquad (3.4.1\text{-}10)$$

The adaptive variance estimator only needs to solve Eqns. (3.4.1-5), (3.4.1-7) and (3.4.1-8) for SC1 model and (3.4.1-5), (3.4.1-9) and

(3.4.1-10) for SC2 model. We finally note that this adaptive scheme maintains the same constant data rate in each DPCM channel as in the nonadaptive scheme of the previous section.

3.4.2 Adaptive Classification

In this method each image column is classified as belonging to one of K predetermined classes. Each of these classes is determined according to the activity in that image column. The activity in an image column is measured by the variance of that column. For a given class of images the probability distribution function of the variances of all the columns can be pre-determined. (Note that this probability distribution function could depend on the size of the image column used in hybrid coding.)

In the DPCM loop, the transformed signal in each DPCM channel is processed as before except that the differential signal is quantized according to the classification of that image column. Thus, the quantizer in the DPCM loop is adapted to the image activity in each column. The number of quantization levels (or bits) depend on the variance of a class so that columns of high dynamic activity are assigned more bits than those of low activity. In this way quantizing bits are employed more efficiently. However, the classification information for each column needs to be transmitted. This can be done by sending an extra $\log_2 K$ bits per column of $\frac{1}{N} \log_2 K$ bits/pixel. Fig. 3.5 shows the adaptive classification scheme. Experimentally it was found that the variance of successive columns, denoted by $\{\xi_j\}$ (say), are highly correlated. Instead of transmitting the classification map, a DPCM loop for transmission of $\{\xi_j^*\}$ are used for classification at the receiver as well as the transmitter. A two bit quantizer was found to give very accurate reproduction of $\{\xi_j\}$.

Let $p(\xi)$ = probability density of the variances ξ of a column.

Suppose the K classes have been predetermined and let (t_k, t_{k+1}), $1 \leq k \leq K$ denote the decision boundaries for the kth class. Since $\xi \geq 0$, $t_0 = 0$, $t_{K+1} = \infty$. Then

$$p_k = \int_{t_k}^{t_{k+1}} p(\xi)d\xi = \text{probability of the kth class} \tag{3.4.2-1}$$

$$\overline{\xi}_k = \frac{1}{p_k}\int_{t_k}^{t_{k+1}} \xi p(\xi)d\xi = \text{variance of the kth class.} \tag{3.4.2-2}$$

First assume that the thresholds $\{t_k\}$ has been somehow predetermined and p_k and $\overline{\xi}_k$ are known. Also, we will assume that for each class k, the image model parameters have been predetermined and are known. To this end, let us denote by $\rho_{i,k}$, $\beta^2(i,k)$ $n_{i,k}$, etc., the quantities for the ith DPCM channel and kth class corresponding to the definitions of ρ_i, $\beta^2(i)$, n_i, etc., respectively. Then, following the development of the foregoing sections, we can write the expected average distortion for any image column (in steady state DPCM) as

$$D = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} \beta^2(i,k)g_{i,k}(n_{i,k})p_k \tag{3.4.2-3}$$

where $n_{i,k} \geq 0$ are such that the expected average bit rate must be constant, i.e.,

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} n_{i,k}p_k = P \ . \tag{3.4.2-4}$$

The bit allocation problem associated with (3.4.2-3) and (3.4.2-4) can be solved along the same lines as in section 3.3. As an example, consider

the case when

$$g_{i,k}(x) = f(x) = 2^{-2x}$$

then one obtains

$$n_{i,k} = \max\left[0, \tfrac{1}{2}\log_2 \frac{\beta^2(i,k)}{\theta}\right] \qquad (3.4.2\text{-}5a)$$

where θ is the solution of

$$\sum_{i=1}^{N}\sum_{k=1}^{K} \max\left[0, \tfrac{1}{2}\log_2 \frac{\beta^2(i,k)}{\theta}\right] p_k = NP \qquad (3.4.2\text{-}5b)$$

and the minimum achievable distortion is

$$D_{min} = \sum_{i=1}^{N}\sum_{k=1}^{K} \min[\theta, \beta^2(i,k)]p_k \; . \qquad (3.4.2\text{-}5c)$$

This method of adaptive hybrid coding requires i) measurement of variance of each column and its transmission by an extra DPCM channel, ii) classification of each column to one of K classes based on thresholds $\{t_k\}$, and iii) storing of K bit allocation tables, one for each class.

3.5 Experimental Results

Several computer experiments have been performed on the Girl and Chemical Plant images to simulate the hybrid coding schemes discussed in the previous sections. Table 3.3 shows the parameters of the various image models used here. In all the experiments, the Cosine transform based semicausal models were used. A 256 × 256 image was divided into 16 strips, each of size 16 × 256. Each strip was hybrid coded independently. This way only a 16 step Cosine transform is needed which makes the scheme

feasible for a real time implementation [46]. In all the performance measurements, the signal to noise ratio (SNR) is defined as

$$\mathrm{SNR} = 20\log_{10}\left(\frac{\text{Peak to Peak Value of the Signal}}{\text{r.m.s. error}}\right) . \qquad (3.5\text{-}1)$$

For an eight bit original data this becomes

$$\mathrm{SNR} = 20\log_{10}\frac{255}{\text{r.m.s. error}} . \qquad (3.5\text{-}2)$$

The normalized mean square error is measured over the entire 256 × 256 image and is defined as

$$\mathrm{NMSE} = \frac{\sum_{i=1}^{256}\sum_{j=1}^{256}(u_{i,j} - u^{*}_{i,j})^2}{\sum_{i=1}^{256}\sum_{j=1}^{256}u^2_{i,j}} . \qquad (3.5\text{-}3)$$

In each DPCM channel a compandor is used as an approximation to the Max quantizer (see Appendix A).

3.5.1 Nonadaptive Hybrid Coding

Figure 3.6 shows the SNR vs. bit rate and rate distortion curve for the SC1, SC2 and the separable covariance image models for the girl image. As expected, the performances of SC2 and separable covariance models are very close. The SC1 model performs better than the other two by about 1.5 to 2 dB, in general, above rates of 1 bit/pixel. The same set of plots for Chemical Plant image is shown in Fig. 3.7. Since the performances of SC2 and separable covariance models are close, for simplicity, the curve for the separable covariance model is not shown in Fig. 3.7.

3.5.2 Adaptive Variance Estimation

Figure 3.8 shows the SNR vs. bit rate and rate distortion curve for the various models of the Girl Iamge for this case. In these experiments, the variance estimates were updated only for channels for which $n_i \geq 2$ bits/pixel.

The performance of SC2 and separable covariance model improves by about 1.5 dB at 1 bit rate and about 4 dB at 2 bit rate. These values can be found from Table 3.4, which gives the quantitative comparisons on a model by model basis. The reason for larger improvment at 2 bit rate is that the variance estimates (based on quantized data) are more accurate since many more channels have $n_i \geq 2$ bits/pixel. The performance improvement for SC1 model is less (only 0.5 dB at 1 bit rate and 1.5 dB at 2 bit rate). This could be because the SC1 model performance was already better (than SC2) to start with and the margin of improvement by updating prediction error variance is small. As the size of the image column (N) is increased, the performance of the adaptive variance estimator was found to improve. This is, again, because the number of channels (for a fixed average rate) with bit allocation $n_i \geq 2$ would increase giving a better estimate of the variances (as well as of the extrapolated values). Similar conclusions can be made for the Chemical Plant Image. The corresponding results are shown in Fig. 3.9 and Table 3.5.

3.5.3 Adaptive Classification

Two different experiments utilizing this method were performed. In the first experiment, a probability distribution function of the column

variances was determined from the histogram of their sample variances. Classification thresholds $\{t_k\}$ were determined for a $K = 4$ case assuming equal probability for each class. Then each column of the image was classified. The actual measurements model for the Cosine transformed vectors was assumed, as

$$v_j(i,k) = \rho_{i,k} v_{j-1}(i,k) + e_j(i,k) \ , \ 1 \leq i \leq N \ , \ k = 1,\ldots 4 \tag{3.5.3-1}$$

with

$$\begin{aligned} Ee_j^2(i,k) &\triangleq \beta^2(i,k) \\ &\triangleq (1-\rho_{i,k}^2)E[v_j(i,k)]^2 \ . \end{aligned} \tag{3.5.3-2}$$

The parameters $\rho_{i,k}$ and $\beta^2(i,k)$ were measured for each class and bit allocation was done according to the method described in the previous chapter. Fig. 3.5 shows the adaptive classification hybrid coding scheme. Fig. 3.10 shows the 16×32 classification map of 16×256 portion of the image with 16 element columns. (Recall that the given 256×256 image was divided into sixteen 16×256, rectangular subimages.)

This method gave a large improvement (see Fig. 3.11 for the Girl Image and Fig. 3.12 for the Chemical Plant Image) in SNR over the previous two schemes (about 5.5 dB at 1 bit rate and 8 dB at 2 bit rate over nonadaptive). However, the above procedure is somewhat impractical for online transmission (could be acceptable for storage) because it requires identification of the $2NK = 8N$ parameters, $\rho_{i,k}$ and $E[v_j(i,k)^2]$ for each image. These parameters are then used to determine the bit allocations for various classifications. Assuming eight bits are required to transmit each of these parameters (this is actually a pessimistic estimate), the additional overhead for their transmission is only

$\frac{8 \times 16 \times 8}{256 \times 256} \simeq \frac{1}{64}$ bit/pixel. The overhead for transmission of the class of the column variance is only 2 bits per column or $\frac{1}{8}$ bit/pixel. Thus, the total overhead is always less than 0.14 bit/pixel. However, the complexity of the scheme at the transmitter is increased substantially.

The complexity of the foregoing method can be reduced considerably by using models for $\rho_{i,k}$ and $\beta^2(i,k)$ according to SC1 or SC2 representations. In our experiments here we assumed

$$\rho_{i,k} = \rho_i \quad \forall\, k \qquad \rho_i = \begin{cases} \gamma/\lambda_{ci}\,, & \text{for SC1} \\ \rho_h\,, & \text{for SC2} \end{cases}$$

$$\beta^2(i,k) = \beta_0^2(i)\sigma_k^2 \quad \forall\, k \qquad \beta_0^2(i) = \begin{cases} \beta_0^2/\lambda_{ci}^2, & \text{for SC1} \\ \beta_0^2/\lambda_{ci}, & \text{for SC2} \end{cases}$$

where σ_k^2 is the variance of the columns belonging to the kth class, $\beta_0^2(i)$ represents the prediction error variance of the ith, unit variance Markov process. Compared to the nonadaptive case, the only additional computation this model requires is the measurement of the sample variance of an image column and classifying it. Figs. 3.13 and 3.14 show the bit allocations for various models used for adaptive classification hybrid coding. Since the first DPCM channel in the Cosine transform represents the d.c. value (or sample mean of the columns), the number of bits allocated to this channel (for a given bit rate and class) were kept the same for all different models. The bit allocation algorithm of previous sections was thus used only for $i = 2,\ldots,N$ in the case of SC1 and SC2 models. Figs. 3.11 and 3.12 show the performance of these models when used for adaptive

classification. Generally, the performance of the SC1 model is superior to that of the SC2. For the Girl Image, it is better than nonadaptive (separable covariance model) hybrid coding by about 3 dB, but is worse than the previous adaptive classification experiment by 2 to 3 dB. This, of course, is the price the designer has to pay for a simplified system. However, for the Chemical Plant image, the adaptive classification scheme does not give much SNR improvement over the non-adaptive scheme, it is worse than the adaptive variance scheme. This is understood by referring to Figs. 3.15 and 3.16, which are the histograms of image column variances. For the Girl image, the variances have a large dynamical range, whereas the variance of the Chemical Plant image are nearly equal. Hence image column activities of the Chemical Plant image remain invariant and therefore classification does not help. Finally, note that this scheme will produce variable bit rate from one image column to the next because of a possible change in classification.

Figs. 3.17-3.19 show the results of various hybrid coding schemes on a model by model basis for the Girl image. Figs. 3.20 and 3.21 are those for the Chemical Plant image. Tables 3.4 and 3.5 give the quantitative comparisons of the above mentioned results. Figs. 3.22 and 3.23 show the encoded Girl image at approximate bit rates of 1 and 2 bits/pixel, and the corresponding figures for the encoded Chemical Plant image are shown in Figs. 3.24 and 3.25. Fig. 3.26 compares the original and the adaptive classified (based on actual measurement model) hybrid encoded images at 1 and 2 bits/pixel.

Finally, observe that as an image column size is increased, the ambiguity in classifying it into different classes will also go up because the activity in a large image column is not likely to be uniform so as to be represented by its variance.

3.6 Extensions of Hybrid Coding

Hybrid coding techniques can be generalized and extended to higher order semicausal models and for coding of noisy images. In this section we briefly discuss the formulations and simulation results.

3.6.1 Hybrid Encoded Image with Channel Error

Since practical communication channels are noisy, it is worthwhile to study the channel noise effects on the Hybrid encoded images. A binary symmetric channel is used here as the noisy communication channel. Fig. 3.27 shows the representation of such a channel, where the probability of receiving an incorrect symbol is P_e and a correct one is $1 - P_e$, regardless of which symbol is transmitted.

Fig. 3.28 shows the effect of channel errors on hybrid coding schemes. Generally, the adaptive classification is quite robust. The nonadaptive and adaptive variance schemes can be made less sensitive to channel errors by reducing somewhat the horizontal correlation parameter. The SC1 model adaptive variance estimation seems to be less sensitive than the SC2 model. Figs. 3.29 and 3.30 show the encoded images in the presence of channel errors. Table 3.6 gives the quantitative comparisons of these results.

3.6.2 Higher Order Predictor in DPCM Loop of Hybrid Coding

This experiment is designed to have a nth order predictor instead of a first order predictor in the DPCM loop. Table 3.7 shows the results of using various predictors. Table 3.8 shows the numerical values of the predictor coefficients which are precalculated on the basis of the statistics of the transformed image strips.

These coefficients, α_i, appear in the nth order autoregressive process (one for each channel)

$$v_j = \sum_{i=1}^{n} \alpha_i v_{j-i} + e_j \tag{3.6.2-1}$$

and are evaluated by solving the linear, Toeplitz equations

$$r_k = \sum_{i=1}^{n} r_{k-i} \alpha_i \tag{3.6.2-2}$$

where $r_k = Ev_j v_{j+k}$ is the covariance of the ith channel process in Hybrid coding.

The variance of the differential signal e_j is

$$\sigma_j^2 = \sigma^2 - \sum_{i=1}^{n} \alpha_i r_i \tag{3.6.2-3}$$

where σ^2 is the variance of the input image.

A block diagram of this scheme is shown in Fig. 3.31 and Table 3.7 shows clearly that first order predictor is a reasonable choice for Hybrid Coding, since the signal to noise ratio improvement is small even though the order of the predictor goes up to four.

3.6.3 Hybrid Coding of Noisy Images

In many image transmission applications, the source image may happen to be noisy. This is because of the image spot pick-up sensor characteristics or due to some inherent limitations inside the mechanism. The foregoing developed coding methods can be modified to encode noisy images, if we know the statistics of noisy images beforehand. Let us assume a given noisy image degraded by additive white Gaussian noise with known signal to noise ratio (S/N) a priori, where S/N is defined as

$$S/N = \frac{\text{Standard deviation of the noise-free signal}}{\text{Standard deviation of the adding noise}} .$$

The additive noise in the given image can be filtered by replacing a predictor in the DPCM loop of Hybrid coder by a discrete Kalman filter. This design allows one to do the joint filtering and coding simultaneously. Fig. 3.32 is the block diagram of this scheme. The message and observation models in the ith channel of Hybrid coder after the transformation are as follows.

<u>Message</u>: $$v_j(i) = \rho_h(i)v_{j-1}(i) + \varepsilon_j(i) \qquad (3.6.3\text{-}1)$$

<u>Observation</u>: $$z_j(i) = v_j(i) + \eta_j(i) \qquad (3.6.3\text{-}2)$$

and

$$E\varepsilon_j(i)\varepsilon_j(i+k) = \sigma^2(1-\rho_h(i))^2\lambda_i\delta_{\ell,0}\delta_{k,0} \qquad (3.6.3\text{-}3)$$

$$E\eta_j(i)\eta_j(i+k) = (S/N)^2\delta_{\ell,0}\delta_{k,0} . \qquad (3.6.3\text{-}4)$$

$\{\lambda_i\}$ are the diagonal entries of eigenvalue of $R_v = \{\rho_v^{|m-n|}\}$, ρ_h and ρ_v are the horizontal and vertical corrections respectively.

The encoding scheme requires first taking the Cosine transform of each noisy image column u_j, then applying a bank of discrete Kalman filter DPCM on each row of the resulting image. Other necessary equations used in this scheme are as follows.

Predicted Observation: $z_{j/j-1}(i) = \rho_h(i)\overline{v}_{j-1}(i)$ (3.6.3-5)

Innovation: $e_j(i) = z_j(i) - z_{j/j-1}(i)$ (3.6.3-6)

Quantizer Output: $e_j^*(i)$ (3.6.3-7)

Current Estimate: $\overline{v}_j(i) = \rho_h(i)\overline{v}_{j-1}(i) + k_j(e)e_j^*(i)$ (3.6.3-8)

The above equations are very similar to the DPCM equations (page 46). Furthermore, $k_j(i)$ appears in (3.6.3-8), is called the Kalman gain and often referred to as a correction factor to the innovation $\hat{e}_j(i)$, (it is $\hat{e}_j^*(i)$ essentially in this scheme) and can be obtained recursively as follows.

Gain: $k_j(i) = \overline{V}_{\tilde{v}_{j/j-1}}(i)[\overline{V}_{\tilde{v}_{j/j-1}}(i) + \overline{V}_{v_j}(i)]^{-1}$ (3.6.3-9)

A priori Variance: $\overline{V}_{\tilde{v}_{j+1/j}}(i) = \rho_h(i)\overline{V}_{\tilde{v}_j}(i)\rho_h(i) + \Psi_\varepsilon$ (3.6.3-10)

A posteriori Variance: $\overline{V}_{\tilde{v}_j}(i) = [1-k_j(i)]\overline{V}_{\tilde{v}_{j/j-1}}(i)$ (3.6.3-11)

where

$$\tilde{v}_{j+1/j}(i) \triangleq v_{j+1}(i) - \overline{v}_{j+1/j}(i) \quad (3.6.3\text{-}12)$$

$$\tilde{v}_j(i) \triangleq v_j(i) - \overline{v}_j(i) \quad (3.6.3\text{-}13)$$

$$\overline{V}_{\tilde{v}_{j+1/j}}(i) = \text{var } v_{j+1/j}(i) \qquad (3.6.3\text{-}14)$$

$$\overline{V}_{\tilde{v}_j}(i) = \text{var } v_j(i) \qquad (3.6.3\text{-}15)$$

For easy reference, the eqns (3.6.3-5) through (3.6.3-15) are summarized in Table 3.9

Since we have a unity variance fixed level quantizer in the DPCM loop, it is required to know the variances of the innovations $e_j(i)$ beforehand. These can be calculated approximately by summing the variances of $\varepsilon_j(i)$ and $\eta_j(i)$. A proper choice of initial conditions $\overline{v}_{0/0}(i)$ and $\overline{V}_{\tilde{v}_{0/0}}(i)$ is required. For simplicity, both are assumed to be zero. This means the initial value of ith DPCM channel is available to the receiver without distortion.

Three different values of S/N of noisy image were simulated in this study. Fig. 3.33 shows the restored images of implementing this scheme in a noisy Girl image. Table 3.10 tabulates the corresponding experimental results. The first entry of each category gives the improvement of signal to noise ratio in decibels, SNR_1, the second entry gives the actual signal to ratio, SNR_2, of the filtered image. These two quantities are related as follows.

$$SNR_2 \triangleq 10\log_{10}\frac{(\text{Peak to Peak Value of Noise-free Image})^2}{\text{Error Variance of Filtered Image}} = 10\log_{10}\frac{(255)^2}{\sigma_e^2}$$

$$SNR_1 \triangleq 10\log_{10}\frac{\text{Error Variance of Noisy Image}}{\text{Error Variance of Filtered Image}} = 10\log_{10}\frac{\sigma_b^2}{\sigma_e^2}$$

$$SNR_2 = 10\log_{10} \frac{(255)^2}{\sigma_e^2}$$

$$= 10\log_{10} \frac{(255)^2}{\sigma_a^2} + 10\log_{10} \frac{\sigma_a^2}{\sigma_b^2} + 10\log_{10} \frac{\sigma_b^2}{\sigma_e^2}$$

$$= 10\log \frac{(255)^2}{\sigma_a^2} + 20\log_{10} \frac{S}{N} + SNR_1$$

where σ_a^2 is the variance of noise-free Girl image (in our example it equals 1816.21). Therefore, the above equation becomes

$$SNR_2 = 10\log \frac{(255)^2}{1816.21} + 20\log_{10} \frac{S}{N} + SNR_1 \quad . \tag{3.6.3-16}$$

In [34] it has shown that semicausal image models can be used to formulate and solve problems in image restoration. Here various hybrid coding schemes based on these models have been extended to encode noisy images. Due to the knowledge of image models, hybrid coding schemes can be successfully applied to encode both noisy and noiseless images. The results of this section promise the effectiveness of previously developed hybrid coding schemes in image data compression applications.

3.7 Summary of Results and Conclusions

Based on the above experiments and the theory discussed in this chapter, the following conclusions are made.

i) It was found that the performance of conventional nonadaptive hybrid coding can be improved up to 5.5 dB at low bit rates (≈1 bit/pixel) by improved statistical characterization of the image via more accurate models and/or via adapting the parameters of the

chosen model to changes in image statistics. The actual measurements model applied in the adaptive classification coding scheme gave the best results. However, this model requires identification of a large number of parameters. In general, the SC1 model performed better than SC2 and separable covariance models by 2 to 3 dB. As expected from theoretical considerations, the SC2 and separable covariance models performed almost identically.

ii) The adaptive variance estimation technique is useful when it is desired to maintain the performance of a hybrid coder, designed for a nominal statistics, in the face of changing statistics (e.g., the correlation parameter). If the image data statistics is close to the design parameters, then the improvement over nonadaptive scheme is marginal (1 to 1.5 dB at 1 bit/pixel). This technique is effective at relatively higher bit rates $\geq$ 1 bit/pixel and for relatively larger image column sizes. It should be considered not so much for achieving low transmission rates as for achieving higher SNR at moderate rates. The complexity over a nonadaptive scheme is marginal, and it can be hardware implemented very easily. Finally, it maintains a constant bit rate from one image column to the next and its performance in the presence of channel errors is no worse than the nonadaptive schemes.

iii) The adaptive classification scheme is a variable bit rate technique (average bit rate from one column to the next can vary) which is robust (with respect to channel errors) and can yield varying degree of performances depending upon the coder complexity. It is relatively

more effective at low bit rates ($\leq$ 1 bit/pixel) and its performance is degraded as the image column size increases.

iv) The Kalman filter DPCM loop scheme is well suited for designing Hybrid coding of noisy images. As expected, the performance of SC2 and separable covariance models are very close. The SC1 model performs better than the other two by about 1 to 0.5 dB. Three different rates have been considered, viz., the infinite (no quantizer in the DPCM loop), 2 bits/pixel and 1 bit/pixel rates. The two bit rate quantizer gives a restored image with SNR only 0.5 dB less than the infinite rate.

v) The overall Kalman filter DPCM coder performance depends on additive noise vs. quantizer noise (bit rate) considerations. At low signal to noise ratios, the additive noise dominates, and moderate changes in bit rates have only marginal effect on improving the SNR of the encoded signal. (Note that the SNR is bounded by the Kalman filter performance without any quantizer noise). At high signal to noise ratios, the effect of quantization is more visible.

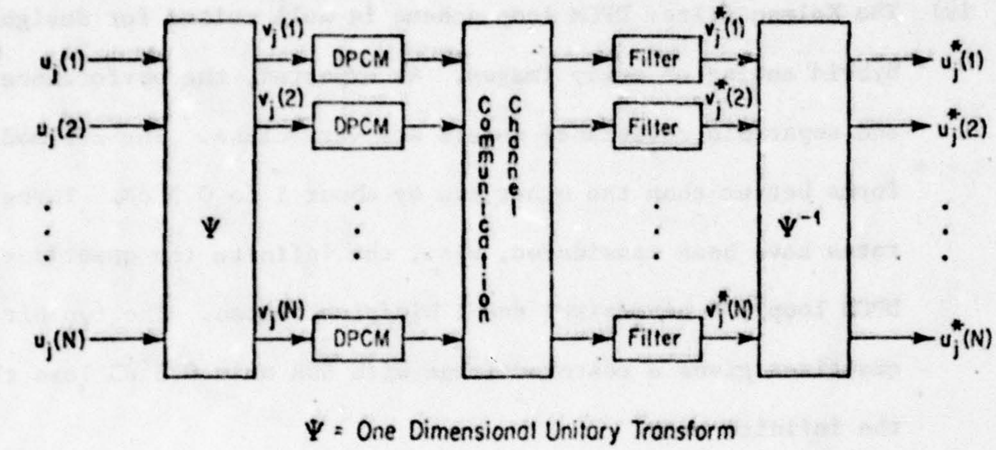


Figure 3.1 Hybrid Coding System

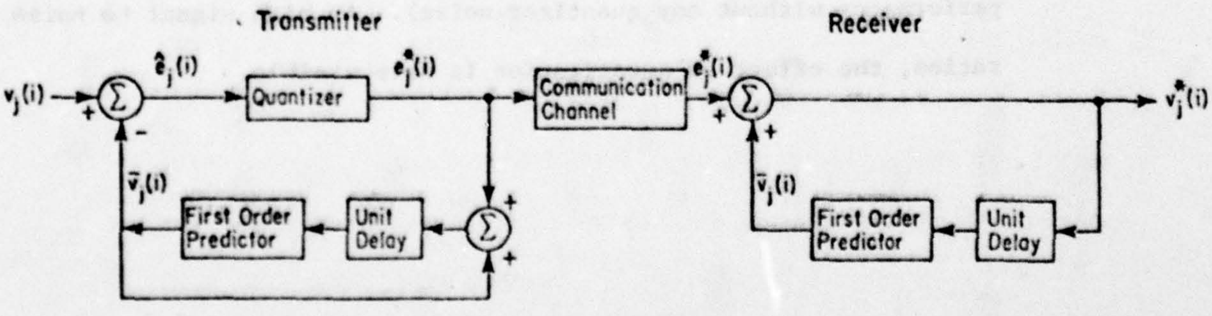


Figure 3.2 The i-th DPCM Channel

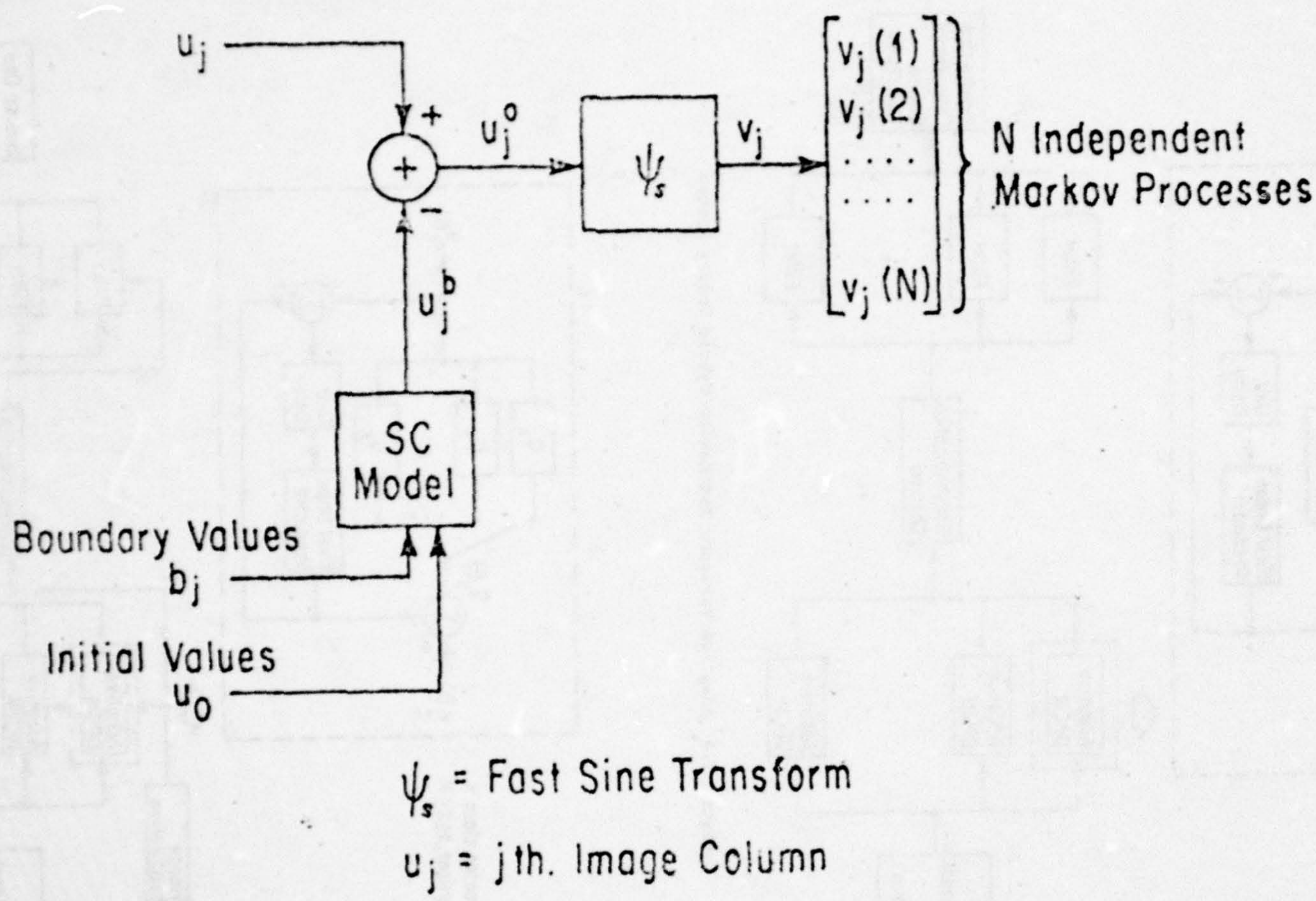


Figure 3.3 Realization of Semicausal Model Decomposition

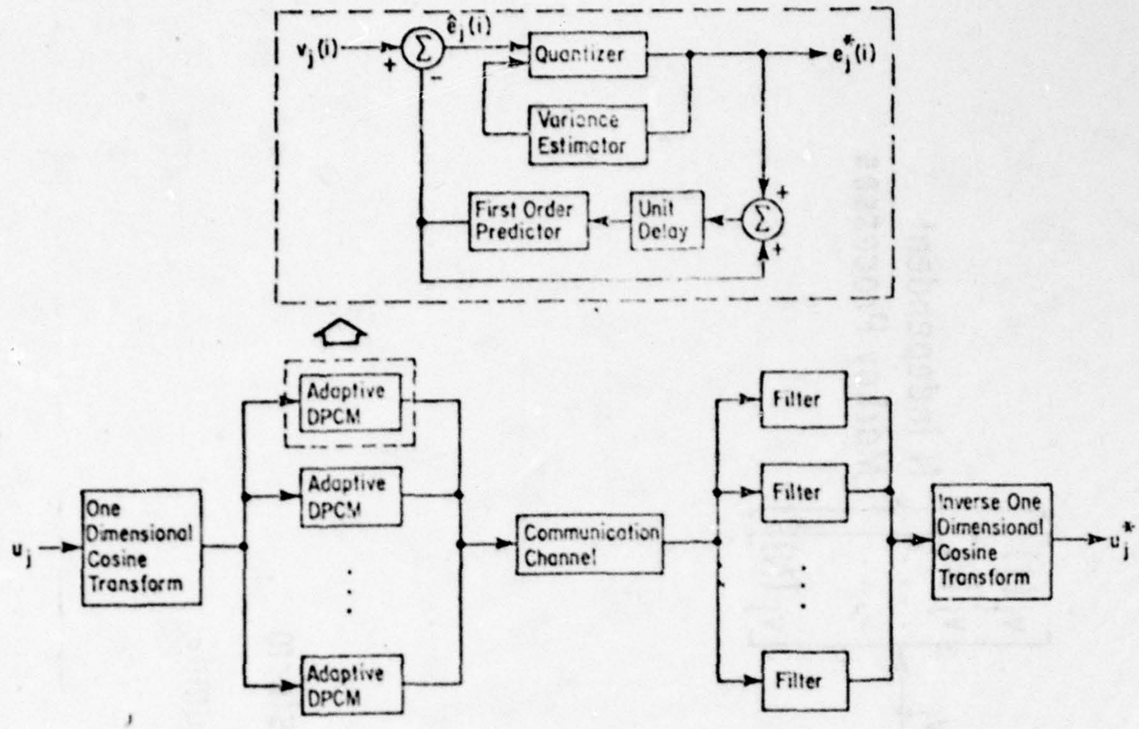


Figure 3.4 Adaptive Variance Estimation Hybrid Coding System

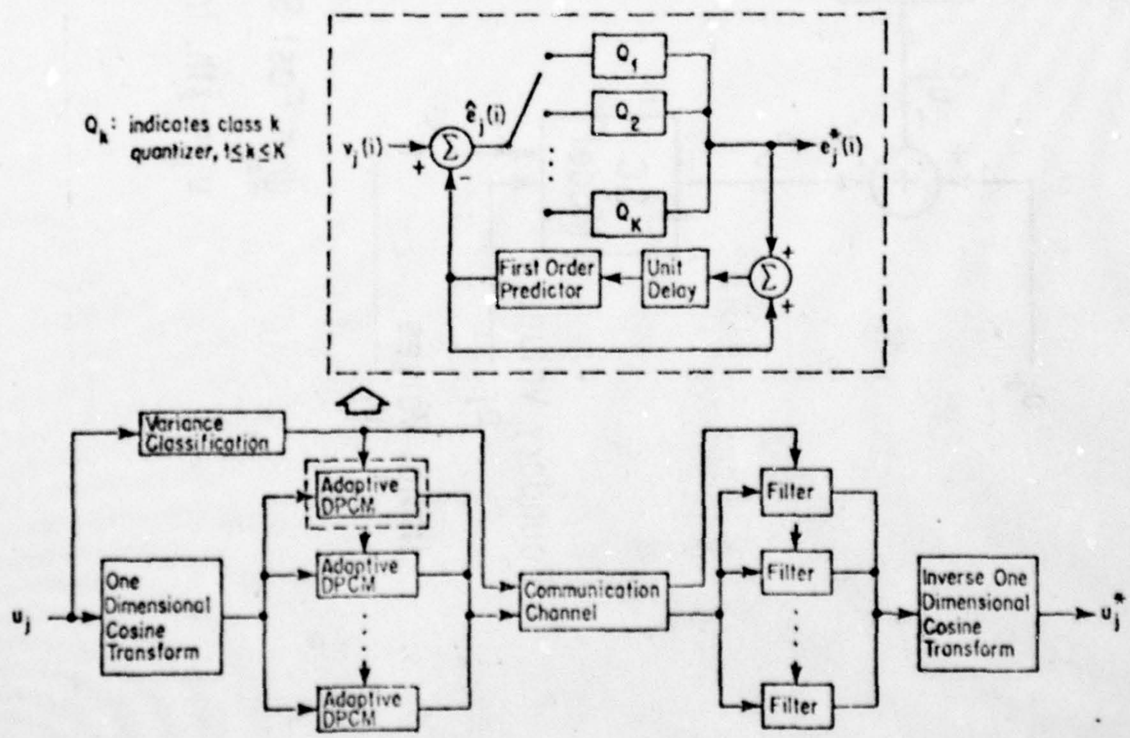


Figure 3.5 Adaptive Classification Hybrid Coding System

AD-A078 781 CALIFORNIA UNIV DAVIS SIGNAL AND IMAGE PROCESSING LAB F/G 9/4
APPLICATIONS OF STOCHASTIC MODELS FOR IMAGE DATA COMPRESSION.(U)
SEP 79 S WANG , A K JAIN DAAG29-78-G-0206

UNCLASSIFIED ARO-16222.3-EL NL

2 OF 3
AD-
A078781

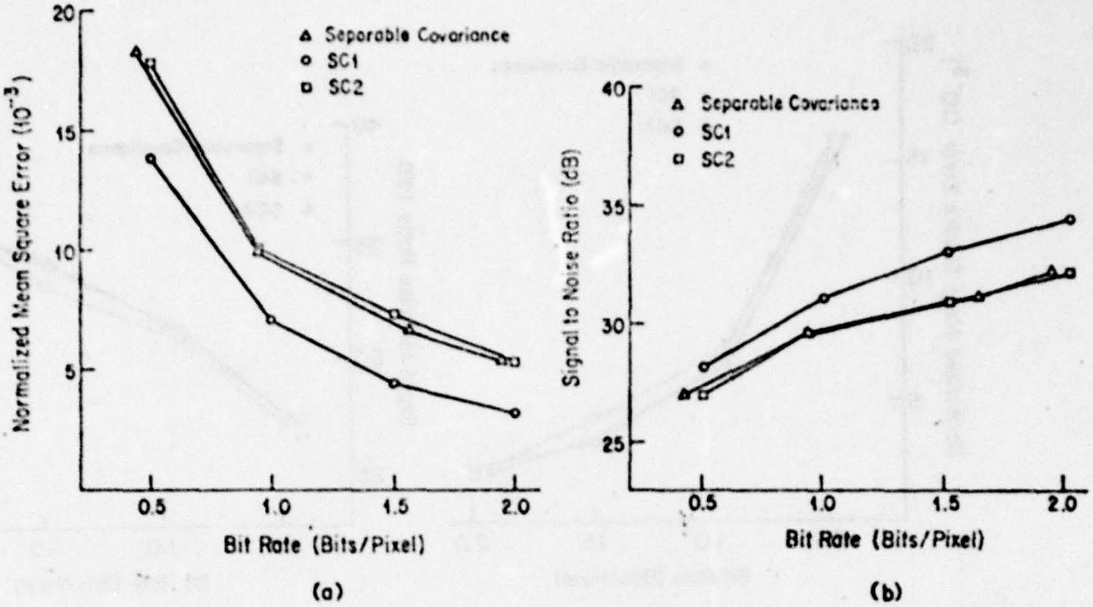


Figure 3.6 Nonadaptive Hybrid Coding Scheme for Various Image Models of the Girl Image. Encoded via 16 × 256 Image Strip, (a) Rate Distortion Curve (b) Signal to Noise Ratio versus Bit Rate

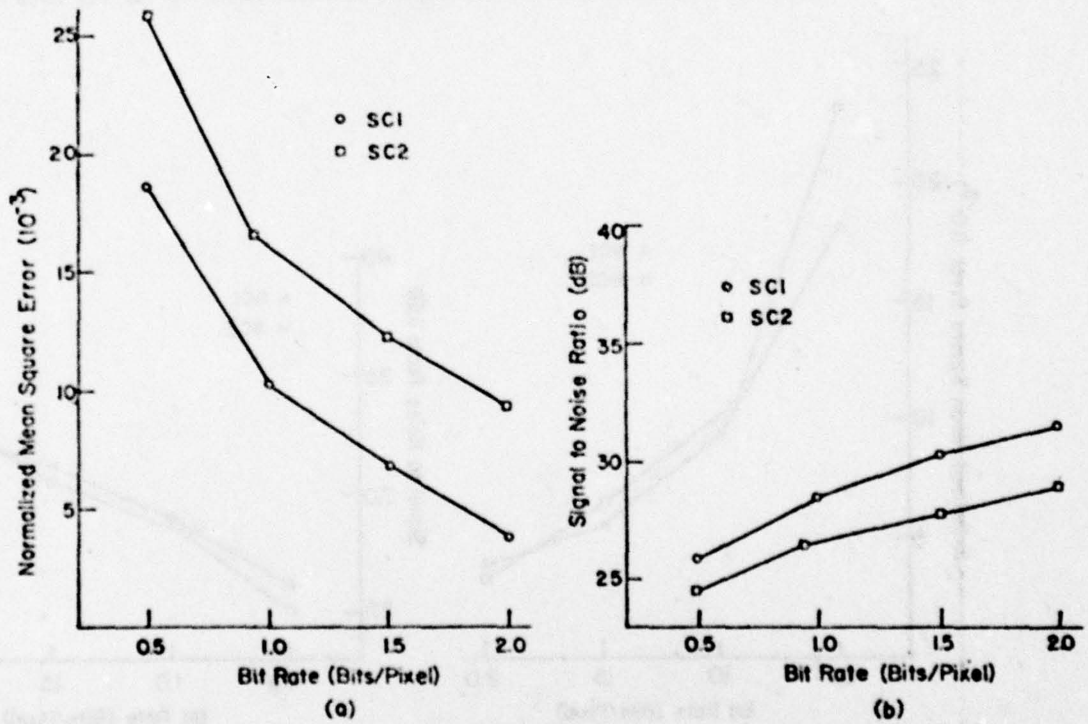


Figure 3.7 Nonadaptive Hybrid Coding Scheme for Various Image Models of the Chemical Plant Image Encoded via 16 × 256 Image Strip, (a) Rate Distortion Curve (b) Signal to Noise Ratio versus Bit Rate

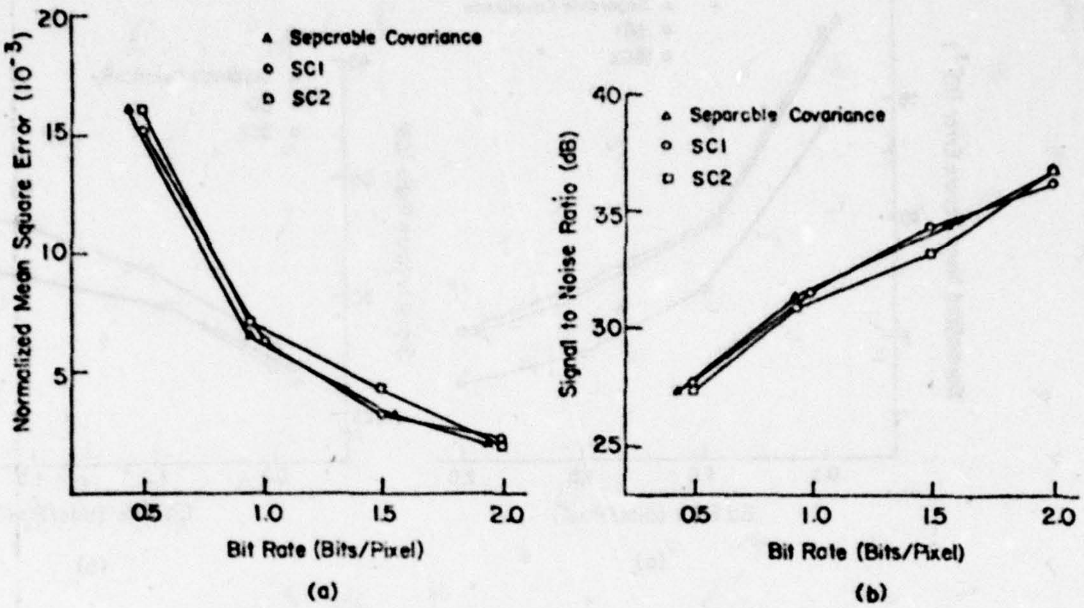


Figure 3.8 Comparison of Adaptive Variance Estimation Hybrid Coding Scheme for Various Image Models of the Girl Image Encoded via 16 × 256 Image Strip. (a) Rate Distortion Curve (b) Signal to Noise Ratio versus Bit Rate

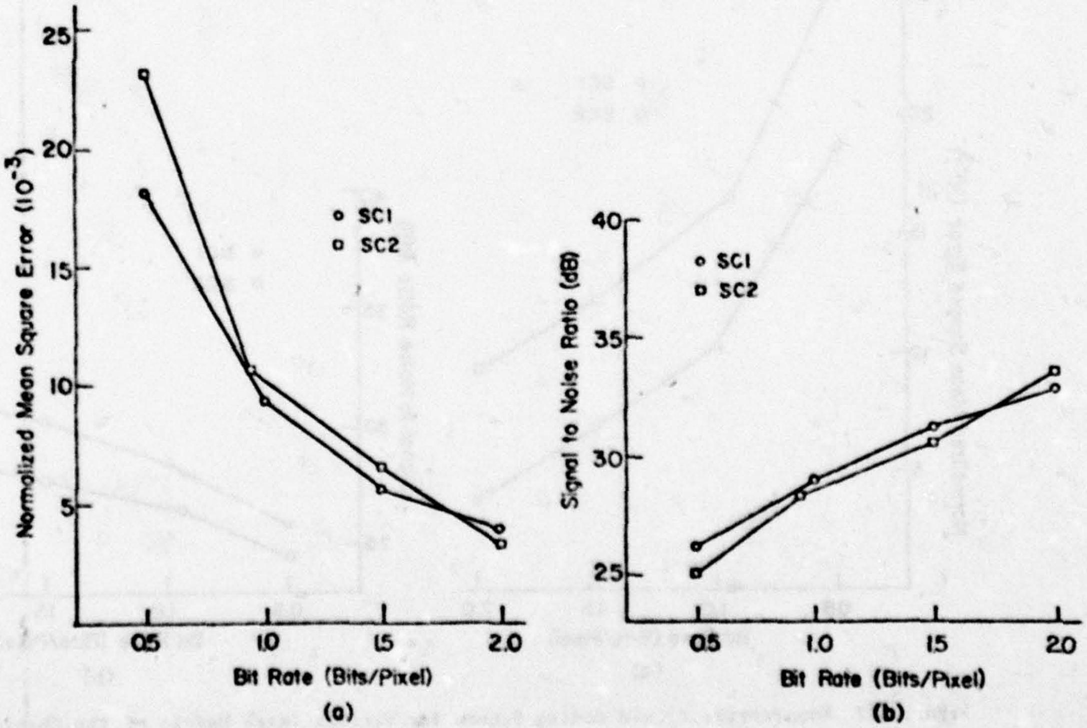


Figure 3.9 Comparison of Adaptive Variance Estimation Hybrid Coding Scheme for Various Image Models of the Chemical Plant Image Encoded via 16 × 256 Image Strip. (a) Rate Distortion Curve (b) Signal to Noise Ratio versus Bit Rate

4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4	4	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	3	4	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	3	4	4	4	4	3	3	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
1	2	2	2	2	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4	4	4	4
1	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	3
2	1	1	1	1	1	1	1	1	1	2	3	3	3	3	2	2	2	3	3	2	2	2	1	1	1	1	1	1	1	2	2
1	1	1	1	1	1	1	1	1	1	1	2	2	3	3	4	4	4	4	3	3	3	2	2	2	2	2	2	2	2	2	2
4	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	4	4
2	2	2	2	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	3	3	3

(a)

2	1	1	1	1	1	1	1	2	3	2	2	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2	2	2	1	1	1	1	1	2	3	3	3	3	3	3	3	4	4	3	4	3	4	4	4	4
3	3	3	2	2	2	2	3	3	3	3	2	1	1	1	2	2	1	1	1	1	1	1	2	1	1	1	1	2	2	2	2
2	2	2	1	2	1	1	1	1	1	1	1	1	1	2	2	2	2	1	1	1	1	1	1	1	1	1	2	2	2	2	2
1	1	1	2	2	3	3	3	3	3	3	2	2	2	2	2	3	3	3	2	2	2	2	2	1	1	2	1	1	1	1	1
3	2	2	2	2	2	2	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	2	2	2	2	3	4	3	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	2	3	3	2	2	2	2	2	2	2	2	2	1	1	1	1
1	1	2	2	2	2	3	4	3	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	3	3	3	3	2	3
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	3	2	3	3	3	3	4	4	3	4	3	2	2	2	3
1	1	1	1	1	1	1	2	2	3	4	4	4	4	2	1	1	2	2	3	4	3	2	2	3	2	1	1	1	2	2	2
3	3	3	4	4	4	4	3	3	3	3	3	2	1	1	1	1	1	1	1	2	2	2	2	2	2	2	3	3	3	3	3
2	2	2	3	3	2	2	1	1	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4
1	1	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
1	1	2	3	3	2	2	2	2	2	2	3	3	2	3	3	3	3	3	3	4	4	4	3	3	4	4	4	4	4	4	3

(b)

Figure 3.10 Classification Map of the Left Quarter Portion 16 × 32 Transformed Sub-Columns of an Image (a) Girl Image (b) Chemical Plant Image, where 1 Indicates the Highest Activity Sub-Columns, 4 Indicates the Lowest Activity Sub-Columns.

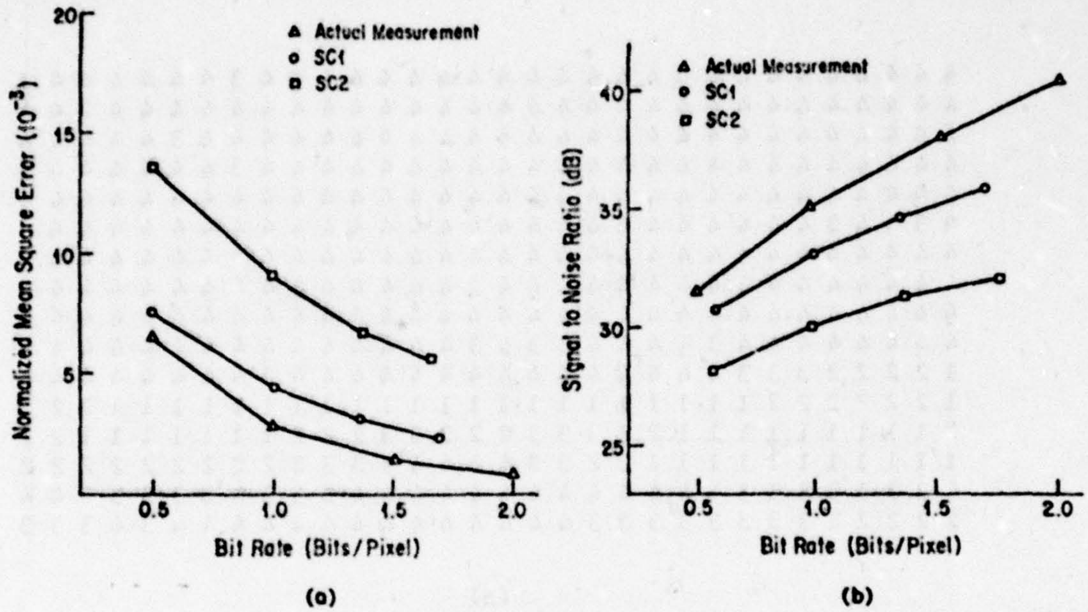


Figure 3.11 Comparison of Adaptive Classification Hybrid Coding Scheme for Various Image Models of the Girl Image Encoded via 16 × 256 Image Stirp, (a) Rate Distortion Curve (b) Signal to Noise Ratio versus Bit Rate

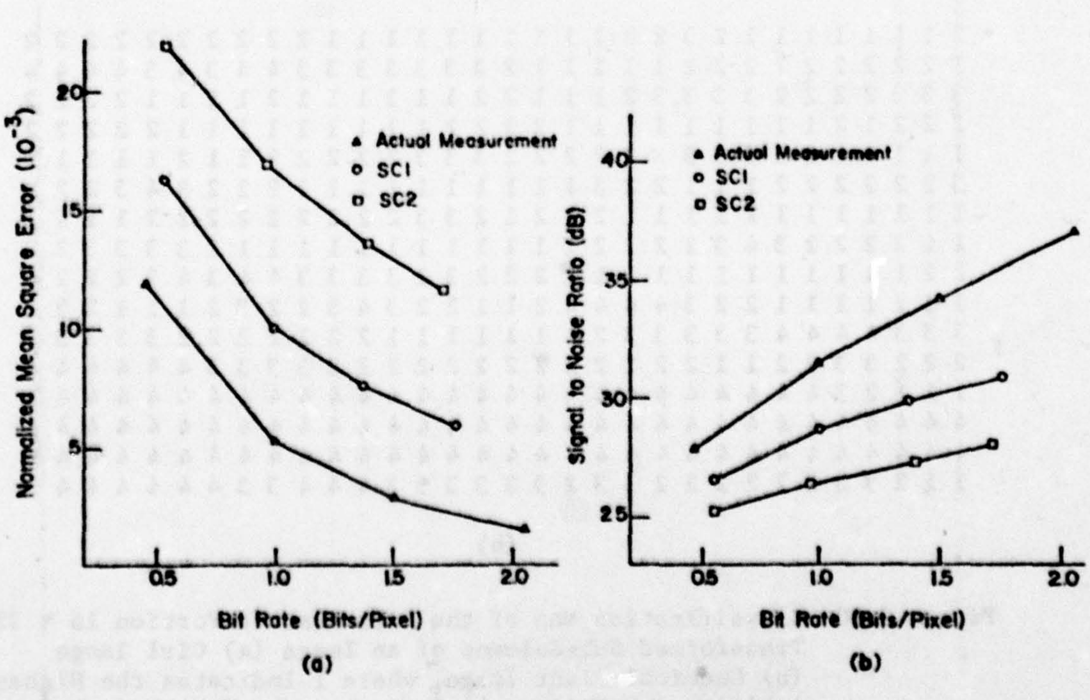


Figure 3.12 Comparison of Adaptive Classification Hybrid Coding Results for Various Image Models of the Chemical Plant Image Encoded via 16 × 256 Image Strip, (a) Rate Distortion Curve (b) Signal to Noise Ratio Versus Bit Rate

I	II	III	IV
5	5	5	5
4	4	3	1
4	3	2	1
4	3	2	1
4	3	2	1
3	3	2	1
3	2	2	1
3	2	1	1
3	2	1	1
3	2	1	1
2	2	1	1
2	2	1	0
2	1	1	1
2	1	1	0
2	1	1	1
2	1	1	0

Average Bit Rate = 2.016

I	II	III	IV
4	5	5	4
4	3	2	1
4	3	2	0
3	3	2	0
3	2	1	0
3	2	1	0
3	2	1	0
2	2	1	0
2	2	1	0
2	1	1	0
2	1	1	0
2	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0

Average Bit Rate = 1.516

I	II	III	IV
4	4	4	3
3	3	2	0
3	2	1	0
3	2	1	0
2	2	1	0
2	2	1	0
2	1	0	0
2	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0

Average Bit Rate = 0.969

I	II	III	IV
3	3	3	3
2	2	1	0
2	1	0	0
2	1	0	0
2	1	0	0
1	1	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Average Bit Rate = 0.500

(a)

I	II	III	IV
5	5	5	5
5	4	3	2
5	4	2	1
5	3	2	1
4	3	2	0
4	2	1	0
3	2	1	0
3	2	0	0
3	1	0	0
3	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	0	0	0

Average Bit Rate = 1.703

I	II	III	IV
4	5	5	4
5	4	2	1
5	3	2	1
4	3	2	0
4	2	1	0
3	2	1	0
3	1	0	0
3	1	0	0
2	1	0	0
2	1	0	0
2	0	0	0
2	0	0	0
2	0	0	0
2	0	0	0
1	0	0	0
1	0	0	0

Average Bit Rate = 1.359

I	II	III	IV
4	4	4	3
4	3	2	0
4	3	1	0
4	2	1	0
3	2	1	0
3	1	0	0
2	1	0	0
2	1	0	0
2	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0

Average Bit Rate = 1.000

I	II	III	IV
3	3	3	3
3	2	1	0
3	2	1	0
3	1	0	0
2	1	0	0
2	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Average Bit Rate = 0.378

(b)

I	II	III	IV
5	5	5	5
6	4	3	2
5	3	2	1
4	3	2	0
4	2	1	0
3	2	1	0
3	2	1	0
3	2	0	0
3	1	0	0
3	1	0	0
3	1	0	0
3	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0

Average Bit Rate = 1.672

I	II	III	IV
4	5	5	4
5	4	3	1
4	3	2	0
4	2	1	0
3	2	1	0
3	2	0	0
3	1	0	0
3	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	0	0	0
2	0	0	0

Average Bit Rate = 1.375

I	II	III	IV
4	4	4	3
5	3	2	1
4	2	1	0
3	2	1	0
3	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	0	0	0
2	0	0	0
2	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0

Average Bit Rate = 0.984

I	II	III	IV
3	3	3	3
4	2	1	0
3	1	0	0
2	1	0	0
2	0	0	0
2	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Average Bit Rate = 0.378

(c)

Figure 3.13 Bit Allocations for Adaptive Classification, K = 4, Girl Image
(a) Actual Measurement Model (b) SC1 Model (c) SC2 Model

I	II	III	IV
4	4	4	4
4	4	3	2
4	3	3	2
4	3	2	2
3	3	2	2
3	3	2	1
3	2	2	1
2	2	2	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	1
2	2	1	0
2	2	1	0

Average Bit Rate = 2.063

I	II	III	IV
4	4	3	3
4	3	2	1
3	3	2	1
3	3	2	1
3	2	2	1
3	2	2	1
2	2	1	1
2	2	1	0
2	1	1	0
2	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0
2	1	1	0
2	1	1	0

Average Bit Rate = 1.516

I	II	III	IV
3	3	3	3
3	2	2	0
3	2	2	0
2	2	1	1
2	2	1	0
2	2	1	0
2	1	1	0
1	1	1	0
1	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0

Average Bit Rate = 1.016

I	II	III	IV
2	2	2	2
2	2	1	0
2	2	1	0
2	1	1	0
1	1	1	0
1	1	0	0
1	1	0	0
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Average Bit Rate = 0.469

(a)

I	II	III	IV
4	4	4	4
5	4	3	3
5	4	3	2
4	3	3	2
4	3	2	1
3	2	2	1
3	2	1	0
3	2	1	0
2	2	1	0
2	1	1	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
1	1	0	0

Average Bit Rate = 1.766

I	II	III	IV
4	4	3	3
5	4	3	2
4	3	3	2
4	3	2	1
3	2	2	1
3	2	1	0
2	2	1	0
2	1	1	0
2	1	0	0
2	1	0	0
2	1	0	0
1	1	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0

Average Bit Rate = 1.375

I	II	III	IV
3	3	3	3
4	3	2	2
4	3	2	1
3	2	2	1
3	2	1	0
2	1	1	0
2	1	0	0
2	1	0	0
1	1	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0

Average Bit Rate = 1.016

I	II	III	IV
2	2	2	2
3	2	2	1
3	2	1	0
2	2	1	0
2	1	0	0
1	1	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Average Bit Rate = 0.547

(b)

I	II	III	IV
4	4	4	4
5	4	4	3
4	3	3	2
4	3	2	1
3	2	2	1
3	2	1	1
3	2	1	0
3	2	1	0
2	2	1	0
2	1	1	0
2	1	1	0
2	1	1	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0

Average Bit Rate = 1.719

I	II	III	IV
4	4	3	3
5	4	3	2
4	3	2	1
3	2	2	1
3	2	1	0
3	2	1	0
2	2	1	0
2	1	1	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
2	1	0	0
1	1	0	0

Average Bit Rate = 1.407

I	II	III	IV
3	3	3	3
4	3	3	2
3	2	2	1
3	2	1	0
2	2	1	0
2	1	0	0
2	1	0	0
2	1	0	0
1	1	0	0
1	1	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0

Average Bit Rate = 0.969

I	II	III	IV
2	2	2	2
4	3	2	1
3	2	1	0
2	1	0	0
2	1	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Average Bit Rate = 0.547

(c)

Figure 3.14 Bit Allocations for Adaptive Classification, K = 4, Chemical Plant Image
(a) Actual Measurement Model (b) SC1 Model (c) SC2 Model

```
MAXIMUM=  .22502E+05 MINIMUM=  .91945E+01 RANGE=  .22353E+05
VARIANCE=  .66196E+07 MEAN=  .10664E+04 STANDARD DEVIATION=  .26113E+04

PSEUDO    LOWER     OCCURRENCES          EACH STAR REPRESENTS   16 OCCURRENCES
LEVEL     LEVEL

                                    160       320       480       640       800
---------------------------------+---------+---------+---------+---------+------
   1   .91945E+01      703 I*******************************************
   2   .35646E+03      766 I***********************************************
   3   .70773E+03      574 I***********************************
   4   .10570E+04      427 I**************************
   5   .14063E+04      319 I*******************
   6   .17555E+04      247 I***************
   7   .21048E+04      162 I**********
   8   .24541E+04      143 I********
   9   .28034E+04      130 I********
  10   .31526E+04      155 I*********
  11   .35019E+04       73 I****
  12   .38512E+04       53 I***
  13   .42004E+04       47 I**
  14   .45497E+04       33 I**
  15   .48990E+04       26 I*
  16   .52482E+04       25 I*
  17   .55975E+04       21 I*
  18   .59468E+04       17 I*
  19   .62961E+04       11 I
  20   .66453E+04       10 I
  21   .69946E+04       12 I
  22   .73439E+04       14 I
  23   .76931E+04        4 I
  24   .80424E+04       12 I
  25   .83917E+04        6 I
  26   .87409E+04        6 I
  27   .90902E+04        7 I
  28   .94395E+04        4 I
  29   .97888E+04        6 I
  30   .10138E+05        3 I
  31   .10487E+05        4 I
  32   .10837E+05        2 I
  33   .11186E+05        4 I
  34   .11535E+05        4 I
  35   .11884E+05        2 I
  36   .12234E+05        2 I
  37   .12583E+05        2 I
  38   .12932E+05        4 I
  39   .13281E+05        1 I
  40   .13631E+05        4 I
  41   .13980E+05       26 I*
  42   .14329E+05        2 I
  43   .14679E+05        4 I
  44   .15028E+05        1 I
  45   .15377E+05        2 I
  46   .15726E+05        1 I
  47   .16076E+05        3 I
  48   .16425E+05        2 I
  49   .16774E+05        2 I
  50   .17123E+05        5 I
  51   .17473E+05        4 I
  52   .17822E+05        4 I
  53   .18171E+05        0 I
  54   .18521E+05        2 I
  55   .18870E+05        2 I
  56   .19219E+05        1 I
  57   .19568E+05        4 I
  58   .19918E+05        3 I
  59   .20267E+05        0 I
  60   .20616E+05        0 I
  61   .20965E+05        1 I
  62   .21315E+05        0 I
  63   .21664E+05        0 I
  64   .22013E+05        0 I
---------------------------------+---------+---------+---------+---------+------
                                    160       320       480       640       800
```

Figure 3.15 Histogram of Image Column Activities (Variances), Girl Image Is Used

MAXIMUM= .10713E+05 MINIMUM= .26970E+02 RANGE= .10684E+05
VARIANCE= .20360E+07 MEAN= .14886E+04 STANDARD DEVIATION= .14270E+04

EACH STAR REPRESENTS 16 OCCURRENCES

(scale: 160 320 480 640 800)

PSEUDO LEVEL	LOWER LEVEL	OCCURRENCES
1	.41945E+01	449
2	.35644E+03	761
3	.70773E+03	718
4	.10570E+04	493
5	.14063E+04	351
6	.17555E+04	411
7	.21048E+04	261
8	.24541E+04	138
9	.28034E+04	205
10	.31526E+04	67
11	.35019E+04	18
12	.38512E+04	14
13	.42004E+04	10
14	.45497E+04	17
15	.48990E+04	13
16	.52482E+04	8
17	.55975E+04	9
18	.59468E+04	8
19	.62961E+04	9
20	.66453E+04	6
21	.69946E+04	21
22	.73439E+04	7
23	.76931E+04	4
24	.80424E+04	4
25	.83917E+04	7
26	.87409E+04	5
27	.90902E+04	8
28	.94395E+04	0
29	.97888E+04	7
30	.10138E+05	4
31	.10487E+05	2
32	.10837E+05	0
33	.11186E+05	0
34	.11535E+05	0
35	.11884E+05	0
36	.12234E+05	0
37	.12583E+05	0
38	.12932E+05	0
39	.13281E+05	0
40	.13631E+05	0
41	.13980E+05	0
42	.14329E+05	0
43	.14679E+05	0
44	.15028E+05	0
45	.15377E+05	0
46	.15726E+05	0
47	.16076E+05	0
48	.16425E+05	0
49	.16774E+05	0
50	.17123E+05	0
51	.17473E+05	0
52	.17822E+05	0
53	.18171E+05	0
54	.18521E+05	0
55	.18870E+05	0
56	.19219E+05	0
57	.19568E+05	0
58	.19918E+05	0
59	.20267E+05	0
60	.20616E+05	0
61	.20965E+05	0
62	.21315E+05	0
63	.21664E+05	0
64	.22013E+05	0

(scale: 160 320 480 640 800)

Figure 3.16 Histogram of Image Column Activities (Variances), Chemical Plant Image Is Used

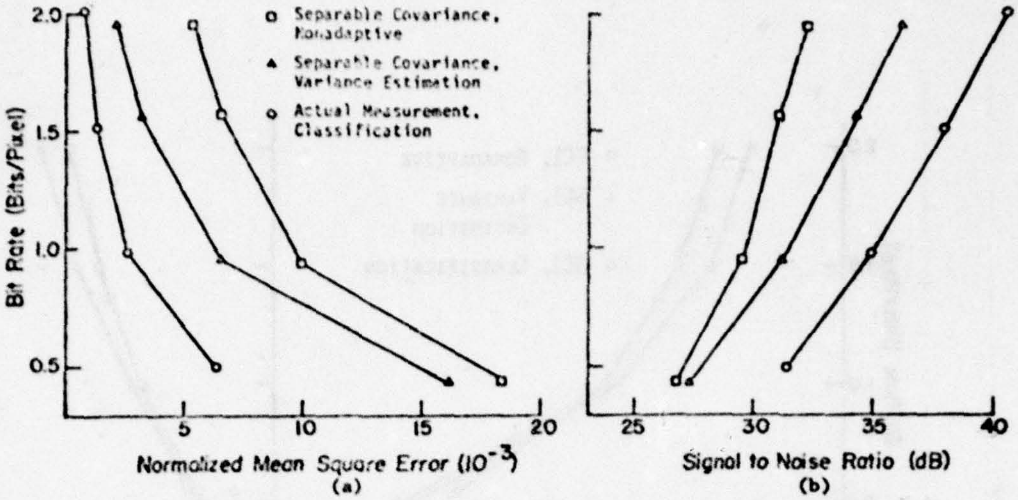


Figure 3.17 Hybrid Coding Results for Separable Covariance Model of the Girl Image Encoded via 16 × 256 Image Strip. (a) Rate Distortion Curve (Bit Rate versus Normalized Mean Square Error) (b) Bit Rate versus Signal to Noise Ratio

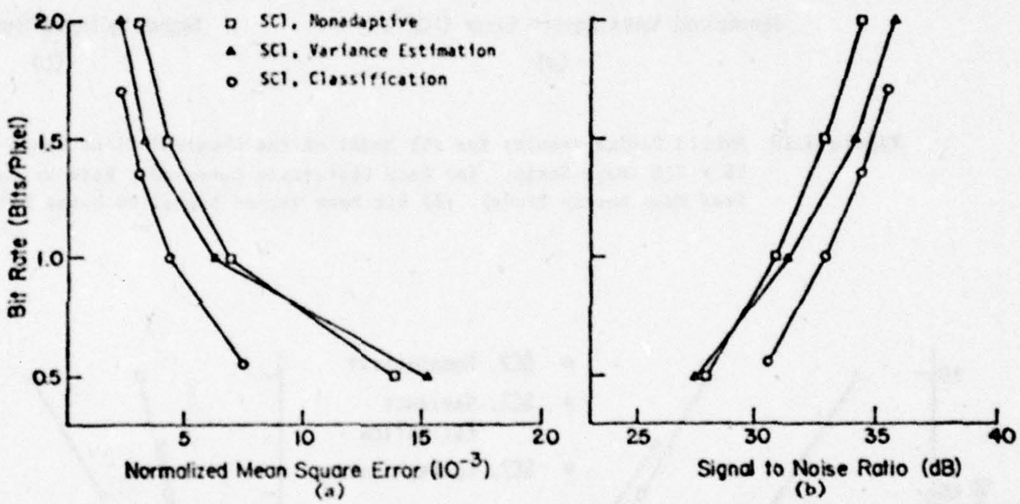


Figure 3.18 Hybrid Coding Results for SC1 Model of the Girl Image Encoded via 16 × 256 Image Strip. (a) Rate Distortion Curve (Bit Rate versus Normalized Mean Square Error) (b) Bit Rate versus Signal to Noise Ratio

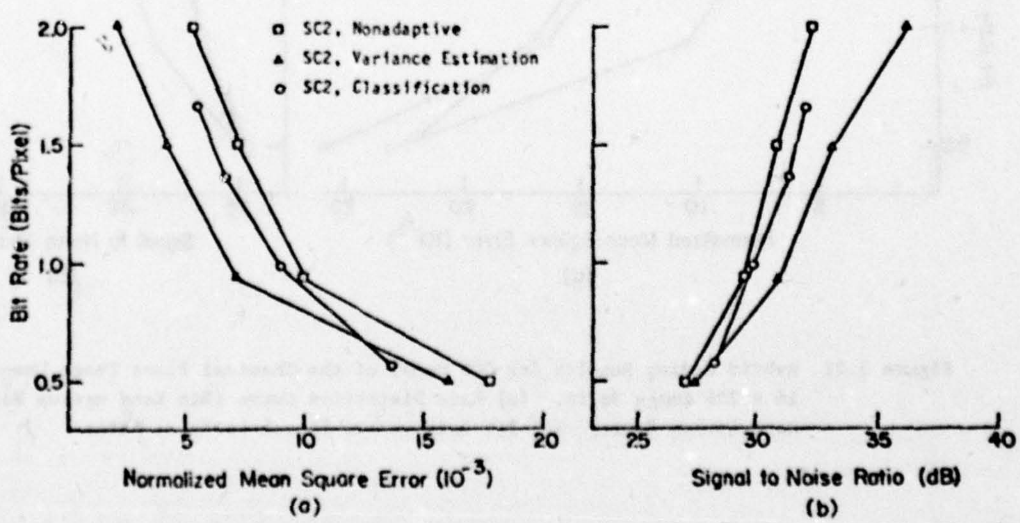


Figure 3.19 Hybrid Coding Results for SC2 Model of the Girl Image Encoded via 16 × 256 Image Strip. (a) Rate Distortion Curve (Bit Rate versus Normalized Mean Square Error) (b) Bit Rate versus Signal to Noise Ratio

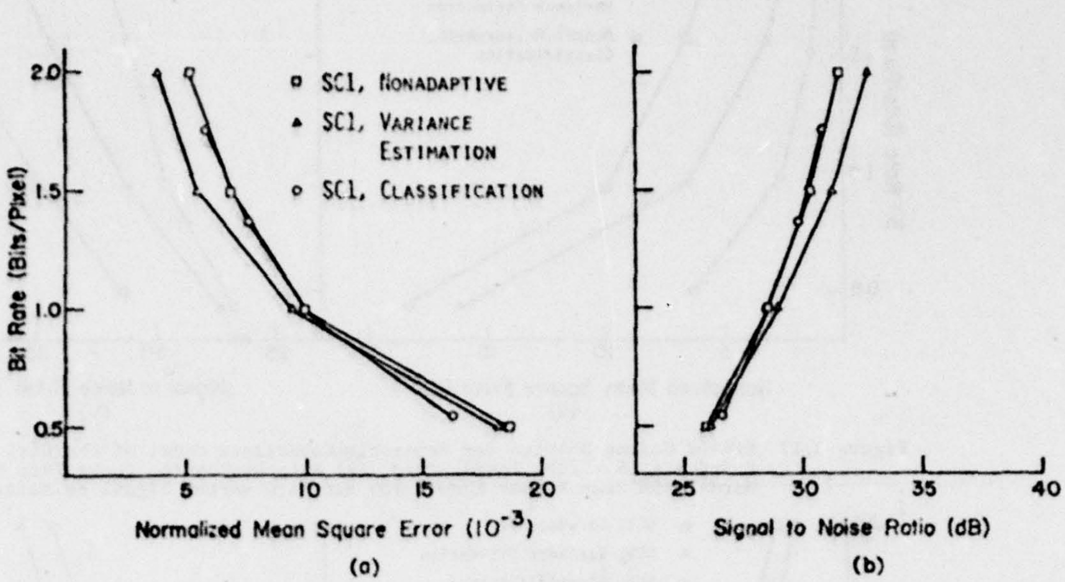


Figure 3.20 Hybrid Coding Results for SC1 Model of the Chemical Plant Image Encoded via 16 × 256 Image Strip, (a) Rate Distortion Curve (Bit Rate versus Normalized Mean Square Error) (b) Bit Rate versus Signal to Noise Ratio

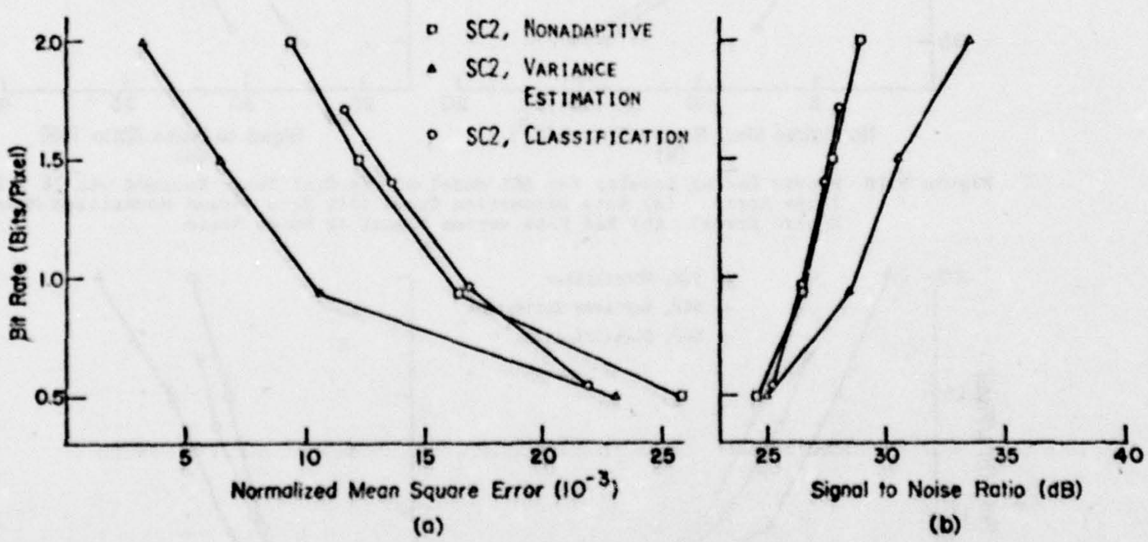


Figure 3.21 Hybrid Coding Results for SC2 Model of the Chemical Plant Image Encoded via 16 × 256 Image Strip, (a) Rate Distortion Curve (Bit Rate versus Normalized Mean Square Error) (b) Bit Rate versus Signal to Noise Ratio

(a) SC1 Model, Nonadaptive

(b) SC1 Model, Adaptive
Variance Estimation

(c) SC1 Model, Adaptive
Classification

(d) SC2 Model, Nonadaptive

(e) SC2 Model, Adaptive
Variance Estimation

(e) SC2 Model, Adaptive
Classification

Figure 3.22 Hybrid Encoded Girl Image at 1 Bit/Pixel Approximately

(a) SC1 Model, Nonadaptive
(2 Bits/Pixel)

(b) SC1 Model, Adaptive Variance
Estimation (2 Bits/Pixel)

(c) SC1 Model, Adaptive Classification (1.7 Bits/Pixel)

(d) SC2 Model, Nonadaptive
(2 Bits/Pixel)

(e) SC2 Model, Adaptive Variance
Estimation (2 Bits/Pixel)

(f) SC2 Model, Adaptive Classification (1.67 Bits/Pixel)

Figure 3.23 Hybrid Encoded Girl Image at 1.7 to 2 Bits/Pixel Approximately

(a) SC1 Model, Nonadaptive

(d) SC2 Model, Nonadaptive

(b) SC1 Model, Adaptive Variance Estimation

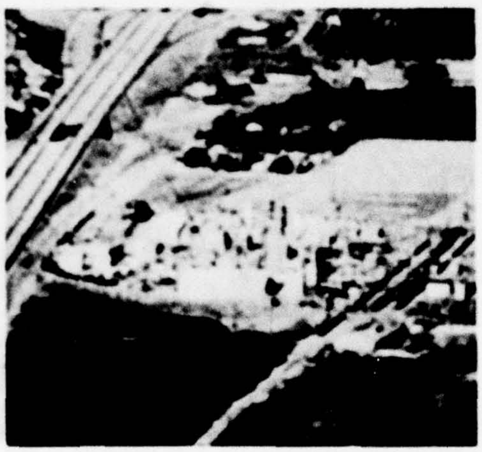
(e) SC2 Model, Adaptive Variance Estimation

(c) SC1 Model, Adaptive Classification

(f) SC2 Model, Adaptive Classification

Figure 3.24 Hybrid Encoded Chemical Plant Image at 1 Bit/Pixel Approximately

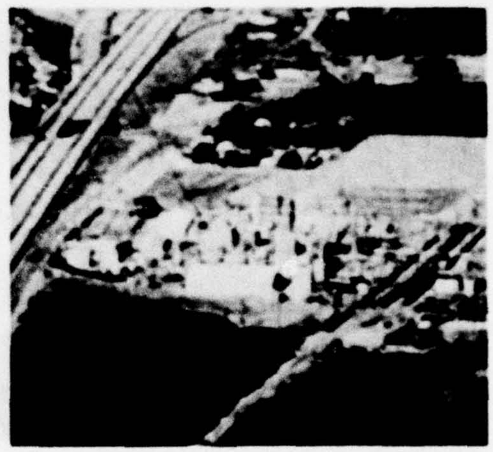
(a) SC1 Model, Nonadaptive
(2 Bits/Pixel)



(b) SC1 Model, Adaptive
Variance Estimation
(2 Bits/Pixel)

(c) SC1 Model, Adaptive Classification (1.77 Bits/Pixel)

(d) SC2 Model, Nonadaptive
(2 Bits/Pixel)



(e) SC2 Model, Adaptive
Variance Estimation
(2 Bits/Pixel)

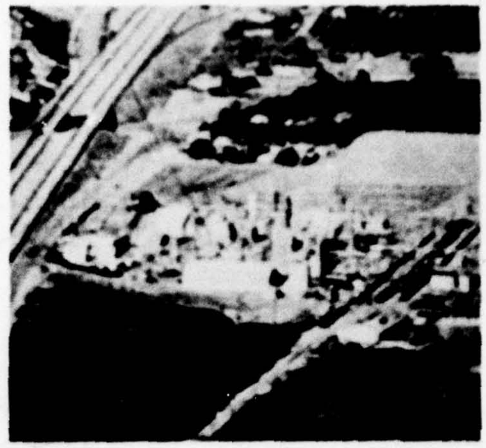
(f) SC2 Model, Adaptive Classification (1.72 Bits/Pixel)

Figure 3.25 Hybrid Encoded Chemical Plant Image at 1.7 to 2 Bits/Pixel Approximately

(a) Original Girl

(b) Actual Measurement Model,
0.97 Bit/Pixel

(c) Actual Measurement Model,
2.02 Bits/Pixel



(d) Original Chemical Plant

(e) Actual Measurement Model,
1.02 Bits/Pixel

(f) Actual Measurement Model,
2.06 Bits/Pixel

Figure 3.26 Actual Measurement Model Hybrid Encoded Image at 1 and 2 Bits/Pixel Approximately

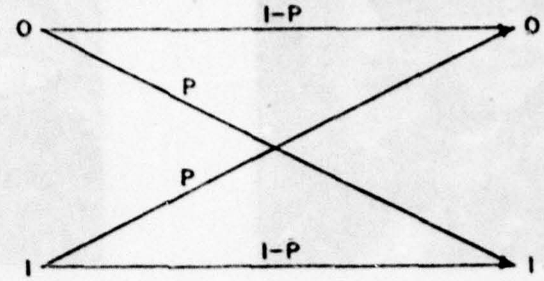


Figure 3.27 Model of a Binary Symmetric Channel

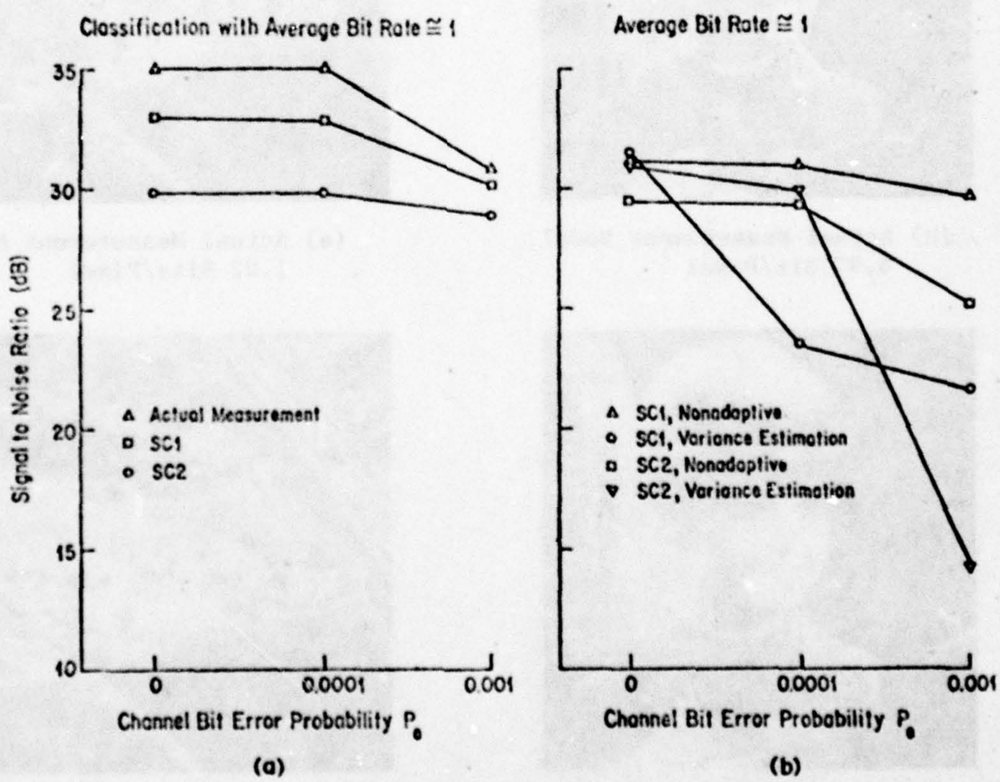


Figure 3.28 Effect of Channel Error

(a) SC1 Model, Nonadaptive

(b) SC1 Model, Adaptive Variance Estimation

(c) SC1 Model, Adaptive Classification

(d) SC2 Model, Nonadaptive

(e) SC2 Model, Adaptive Variance Estimation

(f) SC2 Model, Adaptive Classification

Figure 3.29 Hybrid Encoded Girl Image with Channel Error $P_e = 0.0001$ at 1 Bit/Pixel Approximately

(a) SC1 Model, Nonadaptive

(b) SC1 Model, Adaptive Variance Estimation

(c) SC1 Model, Adaptive Classification

(d) SC2 Model, Nonadaptive

(e) SC2 Model, Adaptive Variance Estimation

(f) SC2 Model, Adaptive Classification

Figure 3.30 Hybrid Encoded Girl Image with Channel Error $P_e = 0.001$ at 1 Bit/Pixel Approximately

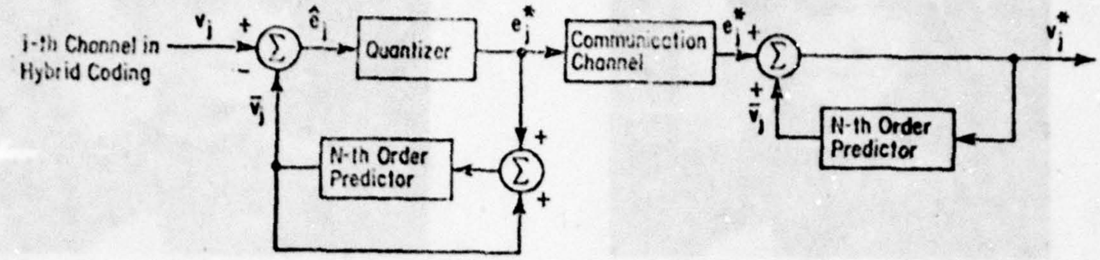


Figure 3.31 Higher Order Predictor and DPCM Channel

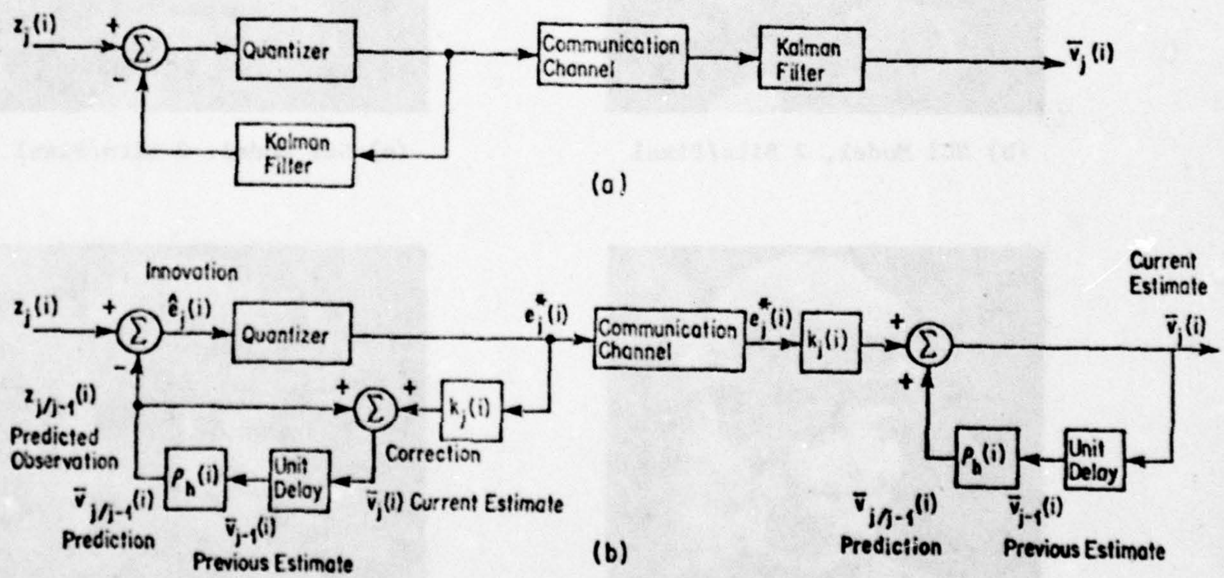


Figure 3.32 Kalman Filter DPCM Loop for the i-th Channel in Hybrid Coding

(a) SC1 Model, Infinite Rate

(d) SC2 Model, Infinite Rate

(b) SC1 Model, 2 Bits/Pixel

(e) SC2 Model, 2 Bits/Pixel

(c) SC1 Model, 1 Bit/Pixel

(f) SC2 Model, 1 Bit/Pixel

Figure 3.33 Restored Images of the Kalman Filter DPCM Loop Scheme Implemented on a Noisy Girl Image (S/N = 5)

Table 3.1 Bit Allocation Patterns Based on Equation (3.3.2-25), Girl Image

Desired Average Bit Rate	2			1.5			1			0.5		
Type of Bit Allocation / DPCM Channel No.	n	[n]	n*	n	[n]	n*	n	[n]	n*	n	[n]	n*
(a) SC1 Model												
1	4.27	4	4	3.77	4	4	3.15	3	3	2.16	2	2
2	4.12	4	4	3.62	4	4	3.00	3	3	2.01	2	2
3	3.74	4	4	3.24	3	3	2.62	3	3	1.63	2	2
4	3.28	3	3	2.78	3	3	2.16	2	2	1.17	1	1
5	2.83	3	3	2.33	2	2	1.70	2	2	0.72	1	1
6	2.42	2	2	1.92	2	2	1.30	1	1	0.31	0	0
7	2.06	2	2	1.56	2	2	0.94	1	1	0	0	0
8	1.75	2	2	1.25	1	1	0.63	1	1	0	0	0
9	1.49	1	1	0.99	1	1	0.37	0	0	0	0	0
10	1.27	1	1	0.77	1	1	0.14	0	0	0	0	0
11	1.08	1	1	0.58	1	1	0	0	0	0	0	0
12	0.93	1	1	0.43	0	0	0	0	0	0	0	0
13	0.81	1	1	0.31	0	0	0	0	0	0	0	0
14	0.71	1	1	0.21	0	0	0	0	0	0	0	0
15	0.65	1	1	0.15	0	0	0	0	0	0	0	0
16	0.61	1	1	0.11	0	0	0	0	0	0	0	0
Average Bit Rate	2.00	2	2	0.15	1.5	1.5	1.00	1.0	1.0	0.50	0.5	0.5
(b) SC2 Model												
1	5.97	6	6	5.30	5	5	4.45	4	4	3.19	3	3
2	3.99	4	4	3.33	3	3	2.51	3	2	1.39	1	1
3	3.05	3	3	2.40	2	2	1.65	2	1	0.77	1	1
4	2.50	3	3	1.89	2	2	1.21	1	1	0.51	1	1
5	2.13	2	2	1.55	2	1	0.95	1	1	0.38	0	1
6	1.87	2	2	1.32	1	1	0.78	1	1	0.29	0	1
7	1.67	2	2	1.15	1	1	0.66	1	1	0.24	0	0
8	1.51	2	2	1.02	1	1	0.57	1	1	0.20	0	0
9	1.39	1	1	0.92	1	1	0.51	1	1	0.17	0	0
10	1.29	1	1	0.85	1	1	0.45	0	1	0.15	0	0
11	1.21	1	1	0.79	1	1	0.43	0	1	0.14	0	0
12	1.15	1	1	0.75	1	1	0.40	0	1	0.12	0	0
13	1.11	1	1	0.71	1	1	0.38	0	0	0.12	0	0
14	1.07	1	1	0.69	1	1	0.36	0	0	0.11	0	0
15	1.05	1	1	0.67	1	1	0.35	0	0	0.10	0	0
16	1.04	1	1	0.66	1	1	0.35	0	0	0.10	0	0
Average Bit Rate	2.00	2	2	1.50	1.56	1.5	1.00	0.94	1	0.50	0.38	0.5

where n: real bit allocation

[n]: indicates the nearest integer of n

n*: integer bit calculated via integer bit allocation algorithm

Table 3.2 Bit Allocation Patterns Based on Equation (3.3.2-21), Girl Image

Desired Average Bit Rate	2			1.5			1			0.5		
Type of Bit Allocation / DPCM Channel No.	n	[n]	n*	n	[n]	n*	n	[n]	n*	n	[n]	n*
(a) Separable Covariance Model												
1	5.39	5	5	4.89	5	5	4.38	4	4	3.44	3	4
2	4.06	4	4	3.56	4	4	3.04	3	3	2.10	2	2
3	3.26	3	3	2.76	3	3	2.25	2	2	1.31	1	1
4	2.70	3	3	2.20	2	2	1.68	2	2	0.74	1	1
5	2.31	2	2	1.81	2	2	1.30	1	1	0.36	0	0
6	2.01	2	2	1.51	2	1	1.00	1	1	0.06	0	0
7	1.78	2	2	1.28	1	1	0.76	1	1	0	0	0
8	1.59	2	2	1.09	1	1	0.57	1	1	0	0	0
9	1.43	1	2	0.93	1	1	0.42	0	1	0	0	0
10	1.31	1	1	0.81	1	1	0.29	0	0	0	0	0
11	1.20	1	1	0.70	1	1	0.18	0	0	0	0	0
12	1.11	1	1	0.62	1	1	0.10	0	0	0	0	0
13	1.05	1	1	0.55	1	1	0.03	0	0	0	0	0
14	1.00	1	1	0.50	0	0	0	0	0	0	0	0
15	0.96	1	1	0.46	0	0	0	0	0	0	0	0
16	0.94	1	1	0.44	0	0	0	0	0	0	0	0
Average Bit Rate	2.10	1.94	2	1.51	1.57	1.5	1.00	0.94	1	0.50	0.44	0.5
(b) SC1 Model												
1	4.27	4	4	3.77	4	4	3.15	3	3	2.16	2	2
2	4.12	4	4	3.62	4	4	3.00	3	3	2.01	2	2
3	3.74	4	4	3.24	3	3	2.62	3	3	1.63	2	2
4	3.28	3	3	2.78	3	3	2.16	2	2	1.17	1	1
5	2.83	3	3	2.33	2	2	1.70	2	2	0.72	1	1
6	2.42	2	2	1.92	2	2	1.30	1	1	0.31	0	0
7	2.06	2	2	1.56	2	2	0.94	1	1	0	0	0
8	1.75	2	2	1.25	1	1	0.63	1	1	0	0	0
9	1.48	1	1	0.99	1	1	0.37	0	0	0	0	0
10	1.27	1	1	0.77	1	1	0.14	0	0	0	0	0
11	1.08	1	1	0.58	1	1	0	0	0	0	0	0
12	0.93	1	1	0.43	0	0	0	0	0	0	0	0
13	0.81	1	1	0.31	0	0	0	0	0	0	0	0
14	0.71	1	1	0.21	0	0	0	0	0	0	0	0
15	0.65	1	1	0.15	0	0	0	0	0	0	0	0
16	0.61	1	1	0.11	0	0	0	0	0	0	0	0
Average Bit Rate	2.00	2	2	1.50	1.5	1.5	1.00	1	1	0.50	0.5	0.5
(c) SC2 Model												
1	6.15	6	6	5.65	6	6	5.12	5	5	4.07	4	4
2	4.17	4	4	3.67	4	4	3.14	3	3	2.08	2	2
3	3.21	3	3	2.71	3	3	2.18	2	2	1.13	1	1
4	2.64	3	3	2.14	2	2	1.61	2	2	0.56	1	1
5	2.24	2	2	1.74	2	2	1.22	1	1	0.16	0	0
6	1.95	2	2	1.45	1	1	0.92	1	1	0	0	0
7	1.71	2	2	1.21	1	1	0.68	1	1	0	0	0
8	1.52	2	2	1.02	1	1	0.49	0	1	0	0	0
9	1.36	1	1	0.86	1	1	0.33	0	0	0	0	0
10	1.23	1	1	0.73	1	1	0.20	0	0	0	0	0
11	1.13	1	1	0.63	1	1	0.10	0	0	0	0	0
12	1.04	1	1	0.54	1	1	0.01	0	0	0	0	0
13	0.98	1	1	0.48	0	0	0	0	0	0	0	0
14	0.93	1	1	0.43	0	0	0	0	0	0	0	0
15	0.89	1	1	0.39	0	0	0	0	0	0	0	0
16	0.87	1	1	0.37	0	0	0	0	0	0	0	0
Average Bit Rate	2.00	2	2	1.50	1.5	1.5	1.00	0.94	1	0.50	0.5	0.5

where n : real bit allocation

[n]: indicates the nearest integer of n

n*: integer bit calculated via integer bit allocation algorithm

Table 3.3 Parameters of Image Models Used in Hybrid Coding

Separable Covariance Model

$$\rho_h = \rho_v = 0.95$$

$$\beta^2(i) = (1-\rho_h^2)/\lambda_i$$

$$\lambda_i = [\Psi R \Psi]_{i,i}$$

Semicausal Model SC1

$\alpha = .4275,\ \gamma = 0.1415,\ \beta^2 = 0.0198$ (Girl Image)

$\alpha = .4275,\ \gamma = 0.1420,\ \beta^2 = 0.0169$ (Chemical Plant Image)

$$\lambda_{ci} = 1 - 2\alpha \cos \frac{(i-1)\pi}{N}, \quad 1 \le i \le N$$

$$\beta^2(i) = \beta^2/\lambda_{ci}^2$$

$$\rho_i = \gamma/\lambda_{ci}$$

Semicausal Model SC2

$$\rho_h = \rho_v = 0.95, \qquad \alpha = \rho_v \Big/ (1+\rho_v^2)$$

$$\lambda_{ci} = 1 - 2\alpha \cos \frac{(i-1)\pi}{N}, \quad 1 \le i \le N$$

$$\beta^2 = (1-\rho_h^2)(1-\rho_v^2) \Big/ (1+\rho_v^2)$$

$$\beta^2(i) = \beta^2/\lambda_{ci}$$

$$\rho_i = \rho_h$$

Actual Measurement Model

$$v_j(i) = \rho_i v_{j-1}(i) + e_j(i)$$

$$\rho_i = E v_j(i) v_{j-1}(i) \simeq \frac{1}{16M} \sum_{j=2}^{M} \sum_{k=1}^{16} v_j^k(i) v_{j-1}^k(i)$$

$$\beta^2(i) = E e_j^2(i) = (1-\rho_i^2) E v_j^2(i) \simeq \frac{(1-\rho_i^2)}{16M} \sum_{j=1}^{M} \sum_{k=1}^{16} [v_j^k(i)]^2$$

where $v_j^k(i)$ = ith element of the jth column of the kth $16 \times M$ size image strip (k=1,...,16 here).

Table 3.4 Hybrid Coding Results of the Girl Image Encoded via 16 × 256 Image Strip

Desired Average Bit Rate	2			1.5			1			0.5		
Method	SNR	N.M.S.E.	Actual Rate	SNR	N.M.S.E.	Actual Rate	SNR	N.M.S.E.	Actual Rate	SNR	N.M.S.E.	Actual Rate
(a) Separable Covariance Model												
Nonadaptive	32.26	0.535%	1.94	31.26	0.674%	1.56	29.57	0.993%	0.94	26.91	1.833%	0.44
Adaptive Variance Estimation	36.24	0.214%	1.94	34.33	0.332%	1.56	31.31	0.666%	0.94	27.47	1.612%	0.44
(b) SC1 Model												
Nonadaptive	34.55	0.316%	2	33.07	0.444%	1.5	31.05	0.707%	1	28.13	1.384%	0.5
Adaptive Variance Estimation	35.97	0.228%	2	34.25	0.338%	1.5	31.48	0.640%	1	27.73	1.518%	0.5
Adaptive Classification	35.80	0.236%	1.70	34.55	0.316%	1.36	33.07	0.444%	1	30.74	0.758%	0.5
(c) SC2 Model												
Nonadaptive	32.30	0.529%	2	30.91	0.730%	1.5	29.53	1.003%	0.94	27.03	1.783%	0.5
Adaptive Variance Estimation	36.25	0.213%	2	33.17	0.434%	1.5	31.00	.715%	0.94	27.48	1.608%	0.5
Adaptive Classification	32.05	0.561%	1.67	31.31	0.666%	1.38	29.97	.906%	0.98	28.20	1.362%	0.5
(d) Actual Measurement Model												
Adaptive Classification	40.59	0.079%	2.02	38.07	0.140%	1.52	35.01	0.284%	0.97	31.42	0.649%	0.5

Tabel 3.5 Hybrid Coding Results of the Chemical Plant Image Encoded via 16 x 256 Image Strip

Desired Average Bit Rate	2			1.5			1			0.5		
Method	SNR	N.M.S.E.	Actual Rate	SNR	N.M.S.E.	Actual Rate	SNR	N.M.S.E.	Actual Rate	SNR	N.M.S.E.	Actual Rate
(a) SC1 Model												
Nonadaptive	31.54	0.520%	2	30.29	0.693%	1.5	28.55	1.034%	1	25.98	1.867%	0.5
Adaptive Variance Estimation	32.82	0.387%	2	31.23	0.558%	1.5	28.98	0.938%	1	26.09	1.820%	0.5
Adaptive Classification	30.93	0.592%	1.77	29.87	0.764%	1.38	28.66	1.009%	1.02	26.58	1.630%	0.55
(b) SC2 Model												
Nonadaptive	28.96	0.941%	2	27.80	1.228%	1.5	26.50	1.658%	0.94	24.56	2.590%	0.5
Adaptive Variance Estimation	33.55	0.327%	2	30.55	0.653%	1.5	28.41	1.069%	0.94	25.05	2.318%	0.5
Adaptive Classification	28.01	1.171%	1.72	27.336	1.368%	1.41	26.40	1.699%	0.97	25.27	2.199%	0.55
(c) Actual Measurement Model												
Adaptive Classification	37.03	0.147%	2.06	34.22	0.280%	1.52	31.48	0.527%	1.02	27.92	1.196%	0.47

Table 3.6 Hybrid Coding Results with Communication Errors, Girl Image Encoded via 16 × 256 Image Strip

Bit Error Probability P_e	0			0.0001			0.001		
Method	SNR	N.M.S.E.	Actual Rate	SNR	N.M.S.E.	Actual Rate	SNR	N.M.S.E.	Actual Rate
(a) Separable Covariance Model									
Nonadaptive	29.57	0.993%	0.94	29.38	1.039%	0.94	27.99	1.430%	0.94
Adaptive Variance Estimation	31.31	0.666%	0.94	28.17	1.373%	0.94	11.77	59.90%	0.94
(b) SC1 Model									
Nonadaptive	31.05	0.707%	1	30.97	0.720%	1	29.60	0.987%	1
Adaptive Variance Estimation	31.48	0.640%	1	23.45	4.064%	1	21.60	6.229%	1
Adaptive Classification	33.07	0.444%	1	32.95	0.456%	1	30.24	0.851%	1
(c) SC2 Model									
Nonadaptive	29.53	1.003%	0.94	29.31	0.105%	0.94	25.17	2.738%	0.94
Adaptive Variance Estimation	31.00	0.715%	0.94	29.95	0.910%	0.94	14.33	33.22%	0.94
Adaptive Classification	29.97	0.906%	0.98	29.97	0.906%	0.98	28.97	1.139%	0.98
(d) Actual Measurement Model									
Adaptive Classification	35.01	0.284%	0.97	35.00	0.285%	0.97	30.88	0.736%	0.97

Table 3.7 The Experimental Results of Higher Order Predictor Implemented on the Girl Image with 16 DPCM Channels

Order of the Predictor	1	2	3	4
Signal to Noise Ratio (dB)	31.202	31.748	31.796	31.792

Table 3.8 Predictor Coefficients of Various Orders for Different Channels in Hybrid Coding

Order of the Predictor	1	2		3			4			
Channel Number	α_1	α_1	α_2	α_1	α_2	α_3	α_1	α_2	α_3	α_4
1	0.989	1.457	-0.473	1.465	-0.498	0.017	1.465	-0.508	0.048	-0.021
2	0.967	1.424	-0.472	1.480	-0.643	0.120	1.486	-0.672	0.188	-0.046
3	0.926	1.313	-0.418	1.369	-0.592	0.133	1.372	-0.605	0.164	-0.023
4	0.866	1.815	-0.368	1.220	-0.483	0.097	1.220	-0.482	0.094	0.003
5	0.851	1.100	-0.292	1.130	-0.406	0.104	1.129	-0.400	0.086	0.016
6	0.820	1.026	-0.251	1.039	-0.305	0.053	1.041	-0.317	0.092	-0.038
7	0.742	0.915	-0.233	0.915	-0.233	-0.001	0.915	-0.240	0.029	-0.033
8	0.666	0.786	-0.179	0.788	-0.191	0.143	0.788	-0.192	0.020	-0.007
9	0.614	0.699	-0.138	0.698	-0.132	-0.009	0.697	-0.133	-0.004	-0.008
10	0.617	0.660	-0.070	0.661	-0.075	0.009	0.660	-0.075	0.004	0.007
11	0.512	0.551	-0.098	0.548	-0.083	-0.027	0.547	-0.084	-0.018	-0.017
12	0.458	0.500	-0.091	0.501	-0.096	0.011	0.501	-0.093	-0.002	0.026
13	0.401	0.413	-0.032	0.414	-0.042	0.025	0.413	-0.040	0.006	0.047
14	0.326	0.337	-0.035	0.339	-0.052	0.050	0.338	-0.051	0.046	0.012
15	0.283	0.284	-0.007	0.284	-0.003	-0.015	0.285	-0.003	-0.024	0.031
16	0.227	0.236	-0.040	0.236	-0.043	0.011	0.236	-0.044	0.014	-0.015

Table 3.9 Discrete Kalman Filter and DPCM Algorithms

Message Model	$v_{j+1}(i) = \rho_h(i)v_j(i) + \epsilon_{j+1}(i)$
Observation Model	$z_j(i) = v_j(i) + \eta_j(i)$
Prior Statistics	$E\epsilon_j(i) = 0,\ E\eta(i) = 0,\ Ev_0(i) = \mu_{v_0}(i) = 0$ $E\eta_j(i)\eta_{j+\ell}(i+k) = (S/N)^2\delta_{\ell,0}\delta_{k,0} = \Psi_\eta\delta_{\ell,0}\delta_{k,0}$ $E\epsilon_j(i)\epsilon_{j+\ell}(i+k) = \sigma^2(1-\rho_h(i))\lambda_i\delta_{\ell,0}\delta_{k,0} = \Psi_\epsilon\delta_{\ell,0}\delta_{k,0}$ $E\epsilon_j(i)\eta_{j+\ell}(i+k) = E\epsilon_j(i)v_0(i+k) = E\eta_j(i)v_0(i+k) = 0$ $var[v_0(i)] = V_{v_0}(i)$
Innovation	$\hat{e}_j(i) = z_j(i) - \rho_h(i)\bar{v}_{j-1}(i)$
Filter	$\bar{v}_j(i) = \rho_h(i)\bar{v}_{j-1}(i) + k_j(i)e^*_j(i)$
Gain	$k_j(i) = V_{\tilde{v}_{j/j-1}}(i)\left[V_{\tilde{v}_{j/j-1}}(i) + V_{v_j}(i)\right]^{-1}$
A Priori Variance	$V_{\tilde{v}_{j+1/j}} = \rho_h(i)V_{\tilde{v}_j}(i)\rho_h(i) + \Psi_\epsilon$
A Posteriori Variance	$V_{\tilde{v}_j}(i) = [1-k_j(i)]V_{\tilde{v}_{j/j-1}}(i)$
Initial Conditions	$\hat{v}_{0/0}(i) = \hat{v}_0(i) = \mu_{v_0}(i) = 0$ $V_{\tilde{v}_{0/0}}(i) = V_{\tilde{v}_0}(i) = V_{v_0}(i)$

Table 3.10 The Experimental Results of Kalman Filter DPCM Loop Implemented on the Noisy Girl Image. The First Entry of Each Category Shows the Improvement of Signal to Noise Ratio in Decibels and the Second Entry Shows Signal to Noise Ratio in Decibels of the Noisy Encoded Image.

Image Model	Kalman Filter DPCM Loop with Quantizer at Average Rate = (.) Bits/Pixel, Where (.) Is				
	∞*	2	1.5	1	0.5
(a) S/N = 1					
Separable Covariance Model	9.1058 24.6652	8.3765 23.9359	8.4023 23.9617	7.5003 23.0597	7.5231 23.0825
Semicausal Model SC1	9.1327 24.6921	8.4537 24.0131	8.4517 24.0111	8.1968 23.7562	8.1660 23.7454
Semicausal Model SC2	8.7611 24.3205	8.1918 23.7512	8.2143 23.7737	7.4203 22.9802	7.4409 23.0003
(b) S/N = 2					
Separable Covariance Model	5.8183 27.3983	4.5647 26.1447	4.5878 26.1678	3.8615 25.4415	3.8129 25.3929
Semicausal Model SC1	5.8895 27.4695	5.2081 26.7881	5.0295 26.6095	4.6383 26.2183	4.0781 25.6581
Semicausal Model SC2	5.7026 27.2826	4.6184 26.1984	4.6257 26.2057	3.9068 25.4868	3.7119 25.2919
(c) S/N = 5					
Separable Covariance Model	2.2492 31.7880	-0.5675 28.9713	-0.9584 28.5804	-1.6039 27.9349	-3.0860 26.4528
Semicausal Model SC1	2.6618 32.2006	1.1673 30.7061	0.7125 30.2513	-0.1349 29.4039	-2.1458 27.3930
Semicausal Model SC2	2.2390 31.7778	-0.5083 29.0305	-1.1176 28.4212	-1.5362 23.0026	-3.3641 26.1747

* : without quantizer in the DPCM loop

CHAPTER FOUR

NONCAUSAL MODELS AND TRANSFORM CODING

4.1 Introduction

Transform coding has attracted considerable attention in the recent years due to its applications in image storage and transmission. This scheme is different from the classical forms of image coding where either the image samples are directly coded (PCM) or the result of a local operation on pixels is coded (DPCM and predictive coding) [9]. In transform coding the image is first unitary transformed and then the transformed samples are quantized and coded for transmission or storage. Fig. 4.1 shows a typical image transform coding system.

Transform coding has some special characteristics, viz.,

(a) Energy Packing: The energy of samples in the transform domain is generally concentrated near the low spatial frequences. This property is exploited to achieve data compression.

(b) Channel Error Immunity: Transform coded images are robust with respect to channel errors. This results from the inherent averaging operation of the transform. Each reconstructed image sample is a weighted function of all transform samples. Hence, the channel error is distributed over all of the reconstructed image samples so that it is less objectionable to a human viewer.

In this chapter, noncausal image representations discussed in Chapter 2 are shown to yield transform coding algorithms. Adaptive transform coding scheme analogous to the classification Hybrid coding method can be designed to account for changes in image statistics.

4.2 Image Transform

Consider an N × N image, $\{u_{i,j},\ i,j = 1,\ldots,N\}$. Let A be a transform operator with elements $\{a_{i,j,k,\ell},\ i,j,k,\ell = 1,\ldots,N\}$ which operates on $\{u_{i,j}\}$ to give

$$v_{k,\ell} = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{i,j,k,\ell} u_{i,j}. \qquad (4.2\text{-}1)$$

The operator A is said to be separable if

$$a_{i,j,k,\ell} = a_{i,k} a_{j,\ell} \qquad (4.2\text{-}2)$$

so that (4.2-1) becomes

$$v_{k,\ell} = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{i,k} a_{j,\ell} u_{i,j}. \qquad (4.2\text{-}3)$$

The above equation can be implemented in two steps. First, a one-dimensional transform is taken along each row (or column) of the image $\{u_{i,j}\}$. Then, a one-dimensional transform is taken along each column (or row) of the resulting image. In matrix notation, this becomes

$$V = A\,U\,A^T. \qquad (4.2\text{-}4)$$

Generally, the matrix operator A is further restricted to be unitary, i.e.,

$$A^{-1} = A^{*T} \qquad (4.2\text{-}5)$$

where * denotes complex conjugate and T denotes transpose. This means the image matrix U can be recovered as

$$U = A^{-1} V (A^T)^{-1} = A^{*T} V A^{*} \qquad (4.2\text{-}6)$$

and the Parseval's energy conservation rule

$$\sum_{i=1}^{N} \sum_{j=1}^{N} |u_{i,j}|^2 = \sum_{k=1}^{N} \sum_{\ell=1}^{N} |v_{k,\ell}|^2 \tag{4.2-7}$$

is satisfied. Common examples of transform matrices are the discrete Fourier (F), Sine (Ψ_s), Cosine (Ψ_c), Hadamard (H), Slant (SL), Karhunen Loeve (Ψ), etc. These are defined as follows:

a. <u>Fourier Transform</u> [3]. The Fourier transform matrix is defined as

$$F_{k,\ell} = \frac{1}{\sqrt{N}} \exp\left\{\frac{-2\pi i(k-1)(\ell-1)}{N}\right\} \qquad 1 \le k,\ell \le N \tag{4.2-8}$$

where $i = \sqrt{-1}$, and the discrete Fourier transform of an image is given by

$$v_{m,n} = \frac{1}{N} \sum_{k=1}^{N} \sum_{\ell=1}^{N} u_{k,\ell} \exp\left\{-\frac{2\pi i}{N}[(m-1)(k-1) + (n-1)(\ell-1)]\right\}. \tag{4.2-9a}$$

Its inverse transform is

$$u_{k,\ell} = \frac{1}{N} \sum_{m=1}^{N} \sum_{n=1}^{N} v_{m,n} \exp\left\{\frac{2\pi i}{N}[(m-1)(k-1) + (n-1)(\ell-1)]\right\} . \tag{4.2-9b}$$

b. <u>Sine Transform</u> [27]. The Sine transform matrix Ψ_s is symmetric and is defined as

$$\Psi_{s_{k,\ell}} = \sqrt{\frac{2}{N+1}} \sin\frac{k\ell\pi}{N+1} \qquad 1 \le k,\ell \le N . \tag{4.2-10}$$

It has been shown [30], if Q_s is an arbitrary, symmetric, tridiagonal, Toeplitz matrix

$$Q_s = \begin{bmatrix} a & b & & 0 \\ b & \ddots & \ddots & \\ & \ddots & \ddots & b \\ 0 & & b & a \end{bmatrix} \qquad (4.2\text{-}10a)$$

then Ψ_s contains the eigenvectors of Q_s and satisfies the relation

$$Q_s \Psi_s = \Psi_s \Lambda_s \qquad (4.2\text{-}10b)$$

where Λ_s is a diagonal matrix whose entries are the eigenvalues of Q_s, given by

$$\lambda_{s_i} = a + 2b \cos\frac{i\pi}{N+1} \qquad 1 \le i \le N\ . \qquad (4.2\text{-}10c)$$

c. Cosine Transform [1]. The Cosine transform matrix Ψ_c is defined as

$$\Psi_{c_{k,\ell}} = \begin{cases} \dfrac{1}{\sqrt{N}}\,, & k = 1,\ 1 \le \ell \le N \\ \sqrt{\dfrac{2}{N}} \cos\dfrac{(2\ell-1)(k-1)\pi}{2N}\,, & 2 \le k \le N\,,\ 1 \le \ell \le N\ . \end{cases} \qquad (4.2\text{-}11)$$

It has been shown [29], the Cosine transform matrix contains the eigenvectors of an arbitrary symmetric tridiagonal matrix Q_c of order N.

$$Q_c = \begin{bmatrix} a-b & b & & & 0 \\ b & a & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & a & b \\ 0 & & & b & a-b \end{bmatrix} \qquad (4.2\text{-}11a)$$

and satisfies the relation

$$Q_c \Psi_c = \Psi_c \Lambda_c \qquad (4.2\text{-}11b)$$

where Λ_c is a diagonal matrix whose entries are given by

$$\lambda_{c_i} = a + 2b \cos \frac{(i-1)\pi}{N} \qquad 1 \leq i \leq N . \tag{4.2-11c}$$

d. Hadamard Transform [54]. The Hadamard transform matrix is a square array whose entry takes only binary values ±1 and rows and columns are orthogonal. The lowest size 2 × 2 orthogonal Hadamard matrix is given by

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} . \tag{4.2-12}$$

The construction rule for a Hadamard matrix of size $N = 2^n$, where n is an integer, is simple. Let H_N be a Hadamard matrix of size N, the matrix

$$H_{2N} = \frac{1}{\sqrt{2}} \left[\begin{array}{c|c} H_N & H_N \\ \hline H_N & -H_N \end{array}\right] \tag{4.2-13}$$

is a Hadamard matrix of size 2N.

e. Slant Transform [55]. For matrix size of N = 2, the Slant transform matrix is identical to the Hadamard transform of order two. That is,

$$SL_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} . \tag{4.2-14}$$

The Slant matrix of order N ($N=2^n, n=3,4,5,\cdots$) can be generated in terms of the Slant matrix of order $\frac{N}{2}$ by the recursive relation

$$
SL_N = \frac{1}{\sqrt{2}}\left[\begin{array}{cc|c|cc|c}
1 & 0 & & 1 & 0 & \\
 & & 0 & & & 0 \\
a_N & b_N & & -a_N & b_N & \\
\hline
\multicolumn{2}{c|}{0} & I_{(N/2)-2} & \multicolumn{2}{c|}{0} & I_{(N/2)-2} \\
\hline
0 & 1 & & 0 & -1 & \\
 & & 0 & & & 0 \\
-b_N & a_N & & b_N & a_N & \\
\hline
\multicolumn{2}{c|}{0} & I_{(N/2)-2} & \multicolumn{2}{c|}{0} & -I_{(N/2)-2}
\end{array}\right]
\left[\begin{array}{c|c}
SL_{\frac{N}{2}} & 0 \\
\hline
0 & SL_{\frac{N}{2}}
\end{array}\right] \tag{4.2-15}
$$

where I_k represents a $k \times k$ identity matrix. The constants a_N and b_N may be computed from the recursive relation

$$
\begin{aligned}
a_2 &= 1 \\
b_N &= [1 + 4(a_{N/2})^2]^{-\frac{1}{2}} \\
a_N &= 2b_N a_{N/2} \; .
\end{aligned} \tag{4.2-16}
$$

f. Karhunen Loeve Transform [37,42]. Here, we consider a special transform which depends on the image statistics.

If $\{x_i,\ i=1,\ldots,N\}$ is a one-dimensional first-order Markov process with covariance matrix R, where

$$
R = \begin{bmatrix}
1 & \rho & \rho^2 & \cdot & \rho^{N-1} \\
\rho & 1 & \rho & & \\
\rho^2 & \rho & 1 & & \\
\cdot & & \cdot & & \\
\cdot & & & \cdot & \\
\cdot & & & & \cdot \\
\rho^{N-1} & & & & 1
\end{bmatrix} \tag{4.2-17}
$$

Then the elements of the KL transform matrix Ψ are given by [56] (N = even)

$$\Psi_{i,j} = \sqrt{\frac{2}{N+\gamma_i^2}} \sin\left[\omega_i\left(j - \frac{N+1}{2}\right) + \frac{j\pi}{2}\right] \tag{4.2-18a}$$

where
$$\gamma_i^2 = \frac{1-\rho^2}{1 - 2\rho\cos\omega_i + \rho^2} \tag{4.2-18b}$$

and $\{\omega_i\}$ are the positive roots of the transcendental equation

$$\tan n\omega = -\frac{(1-\rho^2)\sin n\omega}{\cos\omega - 2\rho + \rho^2\cos\omega}\,. \tag{4.2-18c}$$

Due to the non-periodicity of the sine terms in (4.2-18a), there is no fast algorithm available for implementing the KL transform matrix.

In matrix form, Ψ satisfies

$$R\Psi = \Psi\Gamma \tag{4.2-19}$$

where $\Gamma = \{\gamma_i\}$ is the diagonal matrix of eigenvalues of R, which can be calculated via (4.2-18c).

4.3 Stochastic Decoupling of Noncausal Models by Sine Transform [31]

In Chapter 2 we discussed three different noncausal models. Their spatial structures are depicted in Fig. 4.2 together with the boundary elements needed in representation of an N × N square image. Now we show that each model yields a stochastic decomposition by the Sine transform.

4.3.1 NCl Model

From Chapter 2, we can write the NCl model (2.5.3-1) in matrix form, as

$$U\,Q_s + Q_s\,U = \varepsilon + B \tag{4.3.1-1}$$

where U and ε are $N \times N$ matrices, and Q_s is a tridiagonal, symmetric, Toeplitz matrix defined in (4.2-10a) with $a = \frac{1}{2}$, $b = -\alpha$. From (4.2-10c), its eigenvalues are

$$\lambda_{s_i} = \tfrac{1}{2} - 2\alpha \cos \frac{i\pi}{N+1} \qquad 1 \le i \le N\,. \tag{4.3.1-2}$$

The matrix B is a sparse matrix of the form

$$B = \alpha \begin{bmatrix} u_{01}+u_{10} & u_{02} & u_{03} & \cdots & u_{0,N-1} & u_{0,N}+u_{1,N+1} \\ u_{20} & & & & & u_{2,N+1} \\ u_{30} & & & & & u_{3,N+1} \\ \vdots & & & \bigcirc & & \vdots \\ u_{N-1,0} & & & & & u_{N-1,N+1} \\ u_{N+1,1}+u_{N,0} & u_{N+1,2} & u_{N+1,3} & \cdots & u_{N+1,N-1} & u_{N+1,N}+u_{N,N+1} \end{bmatrix} \tag{4.3.1-3}$$

Note that the four corner boundary elements u_{00}, $u_{0,N+1}$, $u_{N+1,0}$ and $u_{N+1,N+1}$ are not needed in B (see Fig. 4.2). Alternatively we write

$$B \in S_1^{+}(u) . \qquad (4.3.1\text{-}4)$$

Let us denote $\overline{A}$ as an $N^2 \times 1$ vector of lexicographic ordered elements of an $N \times N$ matrix A. Then (4.3.1-1) has a Kronecker product form

$$J\overline{U} \triangleq (I \otimes Q_s + Q_s \otimes I)\overline{U} = \overline{\varepsilon} + \overline{B} \quad . \qquad (4.3.1\text{-}5)$$

From (2.5.3-2), the covariance matrix of $\{\varepsilon_{i,j}\}$ (see Table 2.1 under NC1) can be written as

$$R_\varepsilon \triangleq E\,\overline{\varepsilon}\,\overline{\varepsilon}^T = \beta^2(I \otimes Q_{s_1} + Q_{s_1} \otimes I) \triangleq \beta^2 J_1 \qquad (4.3.1\text{-}6)$$

where Q_{s_1} is defined in (4.2-10a) with $a = \frac{1}{2}$, $b = -\alpha_1$. Now from (4.3.1-5) we obtain the decomposition

$$\overline{U} = \overline{U^o} + \overline{U}_b \quad \text{or} \quad U = U^o + U_b \qquad (4.3.1\text{-}7)$$

where $$\overline{U^o} \triangleq J^{-1}\overline{\varepsilon} \; , \; \overline{U}_b \triangleq J^{-1}\overline{B} \quad . \qquad (4.3.1\text{-}7a)$$

It is clear that $\overline{U}_b$ is obtained by passing the boundary elements B through a noncausal filter J^{-1}. Therefore (4.3.1-7) shows that a discrete random field defined by a noncausal representation has a stochastic decomposition in which U_b is a random field completely determined by

---

Let S_k denote the space of all linear boundary elements of a random field $\{x_{i,j}\}$ which is described as

$$S_k(x) = \{\mathcal{L}[x_{1-k,j}, x_{N+k,j}, x_{i,1-k}, x_{i,N+k}] : 1-k \le i,j \le N+k\}$$

where $\mathcal{L}$ denotes any linear operator.

the boundary elements. *Fig. 4.3 shows the realization algorithm for* noncausal model decomposition of (4.3.1-7).

The covariance matrix of $\overline{U}^o$ is obtained as

$$R_o \triangleq E\,\overline{U}^o\overline{U}^{oT} = J^{-1}E[\overline{\varepsilon}\,\overline{\varepsilon}^T]J^{-1} = J^{-1}R_\varepsilon J^{-1} = \beta^2 J^{-1}J_1 J^{-1} \qquad (4.3.1\text{-}8)$$

$$= \beta^2[I \times Q_s + Q_s \otimes I]^{-1}[I \otimes Q_{s_1} + Q_{s_1} \otimes I][I \otimes Q_s + Q_s \otimes I]^{-1}.$$

If we denote Ψ_s to be the Sine transform matrix, then from (4.2-10) it diagonalizes Q_s. It follows that Ψ_s should also reduce J, J_1, R_o, etc. all to their diagonal form. In fact, it is trivial to check that

$$D = (\Psi_s \otimes \Psi_s)\,X\,(\Psi_s \otimes \Psi_s)\ , \quad \text{for } X = J, J_1, R_o, R_\varepsilon \qquad (4.3.1\text{-}9)$$

where D is an $N^2 \times N^2$ diagonal matrix.

Hence $(\Psi_s \otimes \Psi_s)$ is the KL transform of $\overline{U}^o$, i.e.,

$$\overline{V}^o = (\Psi_s \otimes \Psi_s)\overline{U}^o \quad \text{or} \quad V^o = \Psi_s U^o \Psi_s\ . \qquad (4.3.1\text{-}10)$$

Then V^o is the KL transformed matrix and its elements are uncorrelated, i.e.,

$$E\,V^o_{i,j}V^o_{k,\ell} = d_{i,j}\delta_{i,k}\delta_{j,\ell} \qquad 1 \leq i,j \leq N\ . \qquad (4.3.1\text{-}11)$$

Therefore, it is obvious that one can write [7]

$$D = \beta^2(I \otimes \Lambda_s + \Lambda_s \otimes I)^{-1}(I \otimes \Lambda_{s_1} + \Lambda_{s_1} \otimes I)(I \otimes \Lambda_s + \Lambda_s \otimes I)^{-1} \qquad (4.3.1\text{-}12)$$

which gives

$$E\, V^{o\,2}_{i,j} = d_{ij} = \beta^2 \frac{\left[\lambda_{s_{1_i}} + \lambda_{s_{1_j}}\right]}{\left[\lambda_{s_i} + \lambda_{s_j}\right]^2} \qquad 1 \le i,j \le N \qquad (4.3.1\text{-}13)$$

where λ_{s_1} is obtained from (4.3.1-2) with α replaced by α_1.

For a given image the elements $V^o_{i,j}$ can be easily realized via the equation

$$V^o_{i,j} = \frac{e_{i,j}}{\lambda_{s_i} + \lambda_{s_j}} \qquad (4.3.1\text{-}14)$$

where $\qquad e = \Psi_s\, \varepsilon\, \Psi_s\,,\; V^o = \Psi_s U^o \Psi_s. \qquad$ (4.3.1-14a)

Using the representation (2.5.3-1) and boundary matrix $\{B_{i,j}\}$, find the matrix $\{\varepsilon_{i,j}\}$, then from (4.3.1-2) find the values of λ_{s_i} and λ_{s_j}, and compute the values of $e_{i,j}$ via $e = \Psi_s\, \varepsilon\, \Psi_s$. Since Ψ_s is the Sine transform, a fast algorithm can be used in computing all the equations discussed in the above.

4.3.2 NC2 Model

NC2 representation (2.5.3-6) can be written in matrix form as

$$U H + H U + 2 Q_s U Q_s = \varepsilon + B \qquad (4.3.2\text{-}1)$$

where

$$H_{i,j} = \begin{cases} \tfrac{1}{4} + 2\alpha^2\,, & i = j \\ -\alpha\,, & |i-j| = 1 \\ \alpha^2\,, & |i-j| = 2 \\ 0\,, & \text{otherwise} \end{cases} \qquad (4.3.2\text{-}2)$$

and Q_s is a tridiagonal, symmetric, Toeplitz matrix defined in (4.2-10a) with $a = \tfrac{1}{2}$, $b = -\alpha$, and B is an $N \times N$ sparse matrix which contains the boundary elements and is defined in the following equation.

$$
S = \begin{bmatrix}
\begin{array}{l} 2a(u_{01}+u_{10}) \\ -2a^2(u_{00}+u_{02}+u_{20}) \\ -a^2(u_{-1,1}+u_{1,-1}) \end{array} &
\begin{array}{l} 2au_{02} \\ -2a^2(u_{01}+u_{03}) \\ -a^2(u_{-1,2}+u_{10}) \end{array} &
\begin{array}{l} 2au_{03} \\ -2a^2(u_{02}+u_{04}) \\ -a^2u_{-1,3} \end{array} &
\begin{array}{l} 2au_{04} \\ -2a^2(u_{03}+u_{05}) \\ -a^2u_{-1,4} \end{array} &
\cdots &
\begin{array}{l} 2au_{0,N-2} \\ -2a^2(u_{0,N-3}+u_{0,N-1}) \\ -a^2u_{-1,N-2} \end{array} &
\begin{array}{l} 2a(u_{0,N}+u_{1,N+1}) \\ -2a^2(u_{0,N-2}+u_{0,N}) \\ -a^2(u_{-1,N-1}+u_{1,N+1}) \end{array} &
\begin{array}{l} 2a(u_{0,N}+u_{1,N+1}) \\ -2a^2(u_{0,N-1}+u_{0,N+1}+u_{2,N+1}) \\ -a^2(u_{-1,N}+u_{1,N+2}) \end{array} \\
\begin{array}{l} 2au_{20} \\ -2a^2(u_{10}+u_{30}) \\ -a^2(u_{01}+u_{2,-1}) \end{array} &
-a^2(u_{02}+u_{20}) & -a^2u_{03} & -a^2u_{04} & \cdots & -a^2u_{0,N-2} & -a^2(u_{0,N-1}+u_{2,N+1}) &
\begin{array}{l} 2au_{2,N+1} \\ -2a^2(u_{1,N+1}+u_{3,N+1}) \\ -a^2(u_{0,N}+u_{2,N+2}) \end{array} \\
\begin{array}{l} 2au_{30} \\ -2a^2(u_{20}+u_{40}) \\ -a^2u_{3,-1} \end{array} &
-a^2u_{30} & & & & & -a^2u_{3,N+1} &
\begin{array}{l} 2au_{3,N+1} \\ -2a^2(u_{2,N+1}+u_{4,N+1}) \\ -a^2u_{3,N+2} \end{array} \\
\begin{array}{l} 2au_{40} \\ -2a^2(u_{30}+u_{50}) \\ -a^2u_{4,-1} \end{array} &
-a^2u_{40} & & \boxed{\begin{array}{c} \bigcirc \\ (N-2) \times (N-2) \end{array}} & & & -a^2u_{4,N+1} &
\begin{array}{l} 2au_{4,N+1} \\ -2a^2(u_{3,N+1}+u_{5,N+1}) \\ -a^2u_{4,N+2} \end{array} \\
\vdots & \vdots & & & & & \vdots & \vdots \\
\begin{array}{l} 2au_{N-2,0} \\ -2a^2(u_{N-3,0}+u_{N-1,0}) \\ -a^2u_{N-2,-1} \end{array} &
-a^2u_{N-2,0} & & & & & -a^2u_{N-2,N+1} &
\begin{array}{l} 2au_{N-2,N+1} \\ -2a^2(u_{N-3,N+1}+u_{N-1,N+1}) \\ -a^2u_{N-2,N+2} \end{array} \\
\begin{array}{l} 2au_{N-1,0} \\ -2a^2(u_{N-2,0}+u_{N,0}) \\ -a^2(u_{N-1,-1}+u_{N+1,1}) \end{array} &
-a^2(u_{N-1,0}+u_{N+1,2}) & -a^2u_{N+1,3} & -a^2u_{N+1,4} & \cdots & -a^2u_{N+1,N-2} & -a^2(u_{N+1,N-1}+u_{N-1,N+1}) &
\begin{array}{l} 2au_{N-1,N+1} \\ -2a^2(u_{N-2,N+1}+u_{N,N+1}) \\ -a^2(u_{N-1,N+2}+u_{N+1,N}) \end{array} \\
\begin{array}{l} 2a(u_{N+1,1}+u_{N,0}) \\ -2a^2(u_{N-1,0}+u_{N+1,0}+u_{N+1,2}) \\ -a^2(u_{N+2,1}+u_{N,-1}) \end{array} &
\begin{array}{l} 2au_{N+1,2} \\ -2a^2(u_{N+1,1}+u_{N+1,3}) \\ -a^2(u_{N,0}+u_{N+2,2}) \end{array} &
\begin{array}{l} 2au_{N+1,3} \\ -2a^2(u_{N+1,2}+u_{N+1,4}) \\ -a^2u_{N+2,3} \end{array} &
\begin{array}{l} 2au_{N+1,4} \\ -2a^2(u_{N+1,3}+u_{N+1,5}) \\ -a^2u_{N+2,4} \end{array} &
\cdots &
\begin{array}{l} 2au_{N+1,N-2} \\ -2a^2(u_{N+1,N-3}+u_{N+1,N-1}) \\ -a^2u_{N+2,N-2} \end{array} &
\begin{array}{l} 2au_{N+1,N-1} \\ -2a^2(u_{N+1,N-2}+u_{N+1,N}) \\ -a^2(u_{N+2,N-1}+u_{N,N+1}) \end{array} &
\begin{array}{l} 2a(u_{N+1,N}+u_{N,N+1}) \\ -2a^2(u_{N-1,N+1}+u_{N+1,N-1}+u_{N+1,N+1}) \\ -a^2(u_{N+2,N}+u_{N,N+2}) \end{array}
\end{bmatrix}
\tag{4.3.2-3}
$$

An easier notation for B is $B \in S_2(u)$ (see page 114 footnote).

In Kronecker product notation, (4.3.2-1) becomes

$$[I \otimes H + H \otimes I + 2Q_s \otimes Q_s]\overline{U} = \overline{\varepsilon} + \overline{B} \tag{4.3.2-4}$$

$$J\,\overline{U} = \overline{\varepsilon} + \overline{B} \tag{4.3.2-4a}$$

where $J \triangleq [I \otimes H + H \otimes I + 2Q_s \otimes Q_s]$ and the covariance matrix of $\overline{\varepsilon}$ is

$$R_\varepsilon = E\,\overline{\varepsilon}\,\overline{\varepsilon}^T = \beta^2 J_1 \tag{4.3.2-5}$$

where α is replaced by α_1 in J_1 and Eqn. (4.3.2-4) can be rewritten as

$$\overline{U} = \overline{U}^o + \overline{U}_b \tag{4.3.2-6}$$

where $$\overline{U}^o = J^{-1}\overline{\varepsilon} \quad , \overline{U}_b \triangleq J^{-1}\overline{B} \tag{4.3.2-6a}$$

and the covariance matrix of $\overline{U}^o$ is

$$R_0 = E\,\overline{U}^o\overline{U}^{oT} = J^{-1}E[\overline{\varepsilon}\,\overline{\varepsilon}^T]\,J^{-1} = \beta^2 J^{-1}J_1 J^{-1} \ . \tag{4.3.2-7}$$

Now, the matrix R_0 cannot be diagonalized by the Sine transform matrix ψ_s (see (4.2-10)) and the KL transform of U^o has to be the biorthogonal transform which simultaneously reduces H and Q_s matrices to their diagonal form.

It is noted that H can be further decomposed as

$$H = Q_s^2 + H^b \tag{4.3.2-8}$$

where H^b is a sparse matrix of only two nonzero elements

$$H^b = \begin{bmatrix} \alpha^2 & & \\ & \bigcirc & \\ & & \alpha^2 \end{bmatrix} . \tag{4.3.2-9}$$

Recalling the requirement of $|\alpha| < \frac{1}{4}$ for NC2 model (see (2.5.3-6)), we may assume $H^b \simeq [0]$ and a reasonable approximation of H is

$$H \cong Q_s^2 . \tag{4.3.2-10}$$

Substituting (4.3.2-10) into (4.3.2-14a) yields

$$\begin{aligned} J &= I \otimes H + H \otimes I + 2Q_s \otimes Q_s \\ &\cong I \otimes Q_s^2 + Q_s^2 \otimes I + 2Q_s \otimes Q_s \\ &= J' . \end{aligned} \tag{4.3.2-11}$$

If Ψ_s denotes the Sine transform, it is trivial to prove that

$$(\Psi_s \otimes \Psi_s) J' (\Psi_s \otimes \Psi_s) = I \otimes \Lambda_s^2 + \Lambda_s^2 \otimes I + 2\Lambda_s \otimes \Lambda_s \tag{4.3.2-12}$$

where Λ_s is an N × N diagonal matrix with entries defined in (4.3.2-2).

Following the same steps as we developed in the case of NC1 model, the Sine transform is a good approximation to the biorthogonal transformation in the case of NC2 model.

$$E\, {V_{ij}^o}^2 \cong \beta^2 \frac{\left[\lambda_{s_{1_i}} + \lambda_{s_{1_j}}\right]^2}{\left[\lambda_{s_i} + \lambda_{s_j}\right]^4} \tag{4.3.2-13}$$

Furthermore, since Ψ_s is independent of α, a fast Sine transform exists and the values of $V_{i,j}$ can be easily realized.

4.3.3 NC3 Model

NC3 representation is shown in (2.5.3-10). In matrix notation, it becomes

$$Q_s \, U \, Q_s = \varepsilon + B \tag{4.3.3-1}$$

where Q_s is a tridiagonal, symmetric, Toeplitz matrix defined in (4.2-10a) with $a = 1$, $b = -\alpha$, and eigenvalues λ_s as defined in (4.3.1-2) and

$$B = \alpha Q_s B_1 + \alpha B_2 Q_s - \alpha^2 B_3 \tag{4.3.3-2}$$

where

$$B_1 = \begin{bmatrix} u_{01} & u_{02} & \cdots & u_{0,N} \\ \hdashline & & & \\ & \bigcirc & & \\ & & & \\ \hdashline u_{N+1,1} & u_{N+1,2} & \cdots & u_{N+1,N} \end{bmatrix} \tag{4.3.3-3a}$$

$$B_2 = \left[\begin{array}{c:ccc:c} u_{10} & & & & u_{1,N+1} \\ u_{20} & & & & u_{2,N+1} \\ \cdot & & \bigcirc & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ u_{N,0} & & & & u_{N,N+1} \end{array}\right] \tag{4.3.3-3b}$$

$$B_3 = \begin{bmatrix} u_{00} & & & u_{0,N+1} \\ & & & \\ & \bigcirc & & \\ & & & \\ u_{0,N+1} & & & u_{N+1,N+1} \end{bmatrix} . \tag{4.3.3-3c}$$

Thus

$$B \epsilon S_1(u) \qquad (4.3.3\text{-}4)$$

where the definition of S_1 see page 114 footnote.

Rewriting (4.3.3-1) in Kronecker product form, we have

$$(Q_s \otimes Q_s)\overline{U} = J\,\overline{U} = \overline{\varepsilon} + \overline{B} \qquad (4.3.3\text{-}5)$$

where $$J \triangleq Q_s \otimes Q_s \qquad (4.3.2\text{-}6)$$

so that

$$\overline{U} = \overline{U}^o + \overline{U}_b \qquad (4.3.3\text{-}7)$$

where $$\overline{U}^o \triangleq J^{-1}\overline{\varepsilon}\ ,\ \overline{U}_b \triangleq J^{-1}\overline{B}\ . \qquad (4.3.3\text{-}7a)$$

From Table 2-1, it is noted that the covariance matrix of $\overline{\varepsilon}$ can be written as

$$R_\varepsilon = E\,\overline{\varepsilon}\,\overline{\varepsilon}^T = \beta^2(Q_s \otimes Q_s)\ . \qquad (4.3.3\text{-}8)$$

Thus the covariance matrix of $\overline{U}^o$ is

$$R_0 = E\,\overline{U}^o\overline{U}^{oT} = J^{-1}E[\overline{\varepsilon}\,\overline{\varepsilon}^T]\,J^{-1} = \beta^2 J^{-1} J\, J^{-1} = \beta^2 J^{-1}\ . \qquad (4.3.3\text{-}9)$$

Let Ψ_s be the Sine transform independent of α, recalling the mathematical development described for NC1 and NC2 models, it is seen that $(\Psi_s \otimes \Psi_s)$ is the KL transform of $\overline{U}^o$. Let us define

$$\overline{V}^o = (\Psi_s \otimes \Psi_s)\overline{U}^o\text{, which implies } V^o = \Psi_s U^o \Psi_s\ . \qquad (4.3.3\text{-}10)$$

Then

$$E\,V_{i,j}^{o2} = \frac{\beta^2}{\lambda_{s_i}\lambda_{s_j}}\ . \qquad (4.3.3\text{-}11)$$

Thus the above equation is in a form which is easily realizable. For a given image, the matrix $\{B_{i,j}\}$ is first computed and leads to the matrix $\{U^o_{i,j}\}$. Then from (4.3.1-2) the values of λ_i and λ_j are found. A two-dimensional fast Sine transform is performed over $\{u^o_{i,j}\}$. (4.3.3-11) is used to evaluate the variances of each point of the transformed image $\{V^o_{i,j}\}$.

4.4 Stochastic Formulation of Noncausal Models by Cosine Transform

The Sine transform could be used to analyze the noncausal representations as described in the previous section. For certain special conditions on the outmost boundary variables (refer to Fig. 4.2), the resulting representations could also be analyzed by the Cosine transform.

For many image fields, the adjacent elements of a column (or a row) are highly correlated and often the correlation parameter has a value close to unity. Let us first consider the NC1 model.

For a finite $N \times N$ image random field, the boundary elements of NC1 model that need to be specified in (2.5.3-1) satisfy the conditions

$$
\begin{aligned}
u_{o,j} &= u_{1,j} && 1 \leq j \leq N \\
u_{N+1,j} &= u_{N,j} && 1 \leq j \leq N \\
u_{i,0} &= u_{i,1} && 1 \leq i \leq N \\
u_{i,N+1} &= u_{i,N} && 1 \leq j \leq N
\end{aligned}
\qquad . \qquad (4.4\text{-}1)
$$

This means the two outmost boundary rows of four edges of the $(N+2) \times (N+2)$ field are equal. The NC1 image field subject to (4.4-1) is no longer a stationary field. Rewriting (2.5.3-1) in matrix form with the constraint (4.4-1), we have

$$
Q_c U + U Q_c = \varepsilon \qquad (4.4\text{-}2)
$$

where $$Q_c = \begin{bmatrix} \frac{1}{2}-\alpha & -\alpha & & & \bigcirc \\ -\alpha & \frac{1}{2} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \frac{1}{2} & -\alpha \\ \bigcirc & & & -\alpha & \frac{1}{2}-\alpha \end{bmatrix} \tag{4.4-3}$$

and eigenvalues of Q_c are given by

$$\lambda_c = \frac{1}{2}-2\alpha \cos \frac{(i-1)\pi}{N} \qquad 1 \leq i \leq N \ . \tag{4.4-3a}$$

The subscript c indicates Q_c is now diagonalized by the Cosine transform. In Kronecker product form (4.4-2) becomes

$$(I \otimes Q_c + Q_c \otimes I)\overline{U} = \overline{\varepsilon} \ . \tag{4.4-4}$$

Therefore $$\overline{U} = J^{-1}\overline{\varepsilon} \tag{4.4-5}$$

where $$J \triangleq (I \otimes Q_c + Q_c \otimes I) \ . \tag{4.4-5a}$$

The covariance matrix of R is

$$R = E\,\overline{U}\,\overline{U}^T = J^{-1}E[\overline{\varepsilon}\,\overline{\varepsilon}^T]\,J^{-1} = J^{-1}R\,J^{-1} \tag{4.4-6}$$

where $$R = E\,\overline{\varepsilon}\,\overline{\varepsilon}^T \ . \tag{4.4-6a}$$

Eqn. (4.4-5) does not involve a term in $\overline{U}_b$, compare with (4.3.1-7), any more.

Since we have used the boundary conditions of (4.4-1) in deriving (4.4-2), it means that the error signal $\{\varepsilon_{i,j}\}$ along the four boundary edges is no longer a completely arbitrary random variable. For stationary NC1 model, the covariance matrix of $\overline{\varepsilon}$ was defined in (4.3.1-6), where

$$Q_{s_1} = \begin{bmatrix} \frac{1}{2} & -\alpha_1 & & & \bigcirc \\ -\alpha_1 & \frac{1}{2} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \frac{1}{2} & -\alpha_1 \\ \bigcirc & & & -\alpha_1 & \frac{1}{2} \end{bmatrix} . \tag{4.4-7}$$

Imposition of the boundary conditions also implies the Q_{s_1} could be approximated as Q_{c_1} where

$$Q_{s_1} \cong Q_{c_1} = \begin{bmatrix} \frac{1}{2}-\alpha_1 & -\alpha_1 & & & \bigcirc \\ -\alpha_1 & \frac{1}{2} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \frac{1}{2} & -\alpha_1 \\ \bigcirc & & & -\alpha_1 & \frac{1}{2}-\alpha_1 \end{bmatrix} . \tag{4.4-8}$$

The above equation could be diagonalized by Cosine transform, and the following expression is obtained for R_ε

$$R_\varepsilon \cong \beta^2(Q_{c_1} \otimes I + I \otimes Q_{c_1}) = \beta^2 J_1 \tag{4.4-9}$$

where $$J_1 \triangleq (Q_{c_1} \otimes I + I \otimes Q_{c_1}) . \tag{4.4.9a}$$

Substituting (4.4-9) to (4.4-6), yields

$$R \cong \beta^2 J^{-1} J_1 J^{-1} . \tag{4.4-10}$$

Following the same procedure as we did in section 4.3.1, it is easy to show $(\Psi_c \otimes \Psi_c)$, where Ψ_c is the Cosine transform matrix, could reduce J and J_1 to their diagonal forms. Therefore

$$E\,\overline{V}\,\overline{V}^T \cong \beta^2(\Lambda_c \otimes I + I \otimes \Lambda_c)^{-1}(\Lambda_{c_1} \otimes I + I \otimes \Lambda_{c_1})(\Lambda_c \otimes I + I \otimes \Lambda_c)^{-1} \tag{4.4-11}$$

which implies

$$E\,v^2_{i,j} \cong \beta^2 \frac{\left(\lambda_{c_{1_i}} + \lambda_{c_{1_j}}\right)}{\left(\lambda_{c_i} + \lambda_{c_j}\right)^2} \qquad 1 \le i,j \le N \tag{4.4-12}$$

where $\{\lambda_{c_{1_i}}\}$ and $\{\lambda_{c_i}\}$ could be computed from (4.4-3a). It is obvious that the elements $v_{i,j}$ can be quite easily realized and given by

$$v_{i,j} \cong \frac{e_{i,j}}{\lambda_{c_i} + \lambda_{c_j}} \tag{4.4-13}$$

where $\qquad e \cong \Psi_c\, \varepsilon\, \Psi_c\,, \quad V \cong \Psi_c U \Psi_c\,. \qquad$ (4.4-13a)

Once again, fast algorithm can be used to implement the Cosine transform in the above equations.

For NC2 and NC3, similar models can be obtained by simply assuming the outermost boundary values to be equal. Table 4.1 summarizes the formulas for the three models which are useful in the design of transform coders for random fields represented by noncausal models.

4.5 Transform Coder Design for Noncausal Models via Cosine Transform

In coding problems, one would like to find the minimum number of bits required to represent a random field subject to the specific amount of distortion. A common approach of evaluating the performance of a coding scheme is to obtain its distortion vs. rate characteristics. Referring to Fig. 4.1, which shows the overall transform coding system, an image U is unitarily transformed to obtain V. Each element of V is then quantized independently by a zero memory quantizer and is used for

transmission and/or storage. For reconstruction of the original image, one simply performs the inverse transformation of the quantized samples.

One basic property of the KL transform is that it completely decorrelates the transform samples. Other unitary transforms reduce the inter-sample correlation. The transform coder efficiency is maximized when the transformed samples are uncorrelated. The quantizer design depends on the probability distribution of the transformed samples. In our study, we have found the transform sample $v_{i,j}$ can be modeled quite well by the Gaussian density model

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-(x-\mu)^2/2\sigma^2] \qquad (4.5\text{-}1)$$

where μ and σ are the mean and standard deviation of the random variable x. Many times the first sample v_{11} is modeled by a Rayleigh density, because, for many transforms, it is proportional to the average value (or the so called d.c) of the image data, which for non-negative image data would be a non-negative random variable. However, if the image data has been converted to a zero mean data before processing, then the Gaussian density model suffices. In the sequel, without loss of generality, we will assume $\{u_{i,j}\}$ to be zero mean random variables, therefore, $\{v_{m,n}\}$ are also zero mean random variables. Let

$$\sigma^2_{v_{m,n}} \triangleq E\, v^2_{m,n} \qquad (4.5\text{-}2)$$

be the variance of the transformed sample $v_{m,n}$. In Fig. 4.1 if there are no channel errors, then the coder/decoder will generate a perfect transaction and we will have $\hat{v}^*_{m,n} = v^*_{m,n}$. The average mean square distortion in encoding entire N × N image is defined as

$$D = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} E\,(\hat{u}^*_{i,j} - u_{i,j})^2. \qquad (4.5\text{-}3)$$

Since the transformation is unitary, the Parseval's relation implies

$$D = \frac{1}{N^2} \sum_{m=1}^{N} \sum_{n=1}^{N} E\,(\hat{v}^*_{m,n} - v_{m,n})^2 \qquad (4.5\text{-}4)$$

and in the absence of channel errors

$$D = \frac{1}{N^2} \sum_{m=1}^{N} \sum_{n=1}^{N} E\,(v^*_{m,n} - v_{m,n})^2 . \qquad (4.5\text{-}4)$$

Assuming all the quantizers are identical in their characteristics, $f(b)$, where

$f(b)$ = mean square distortion of a quantizer with 2^b levels for unit variance input random variable

p = desired average bit rate in bits/pixel of the image random field

we can write

$$D = \frac{1}{N^2} \sum_{m=1}^{N} \sum_{n=1}^{N} f(b_{m,n})\,\sigma_v^2(m,n) \qquad (4.5\text{-}5)$$

where $b_{m,n}$ = number of bits allocated to the sample $v_{m,n}$. Now, since the total number of bits available is fixed, i.e.,

$$\sum_{m=1}^{N} \sum_{n=1}^{N} b_{m,n} = N^2 p \qquad (4.5\text{-}6)$$

and

$$b_{m,n} \geq 0 \qquad (4.5\text{-}7)$$

the overall distortion given by (4.5-5) is minimized by finding the optimal bit allocation among the various samples subject to the constraints

of (4.5-6) and (4.5-7). Following Chapter 3, section 3.3.2, and Appendix A, the bit allocation problem can be solved as before and the distortion vs. rate characteristics can be determined. For the noncausal models considered here, $\sigma_v^2(m,n)$ are obtained from Table 4.1 for Cosine transform coding. The overall coding system is shown in Fig. 4.4(a). In general when the conditions of (4.4-1) are invalid, boundary variables are arbitrary, we can use the following method.

4.6 Fast KL (Sine) Transform Data Compression

In either noncausal model (NC1, NC2 or NC3), the decomposition $U = U^o + U_b$ plays an important role in the analysis. Fig. 4.4(b) illustrates a Sine transform data compression scheme. First the boundary samples are quantized and the boundary response U_b^* is computed. Subtracting U_b^* from U gives $\tilde{U}^o$, which is nearly equal to U^o, if quantization errors in boundary samples are small. The covariance matrix of U^o is a function of Q for NC1 and NC3 models and a function of Q and H for NC2 model. It has been shown that Sine transform serves as the exact KLT for NC1 and NC3 models and a good approximation to the NC2 model. $\tilde{U}^o$ is therefore quantized with a bank of noncausal model influenced quantizers and transmitted over the channel.

In actual implementation, a large image would be first divided into small blocks and each block is coded independently. Since the noncausal models require separate coding of image boundaries, the number of boundary samples that are needed to be quantized for each block can be minimized by overlapping the boundaries of the various image blocks, namely, only the right vertical and lower horizontal need to be quantized. Rate

distortion calculations for one-dimensional noncausal models (with one overlapping boundary point) have shown to give lower bit rates than the conventional KLT or other unitary transform coding without any overlap. Details of these aspects and comparisons with conventional transform coding techniques are given in [36].

4.7 Experimental Results

Several computer experiments have been performed to simulate the transform coding schemes discussed in the previous sections. In all the experiments only Cosine transform based scheme was implemented. A 256×256 image was divided into 16 image blocks, each of size 64×64. Each block was transform coded independently. The size of image block could be chosen arbitrarily, the size 64×64 being chosen for comparison with the methods of the next chapter.

Figure 4.5 shows a typical bit allocation pattern of a NC1 Cosine transform image. Tables 4.2 and 4.3 summarize the results for the Girl and Chemical Plant images respectively. Figure 4.6 shows rate distortion curve and rate vs. SNR for the Girl image. The corresponding characteristics for the Chemical Plant image are shown in Figure 4.7. Figure 4.8(a)-(c) show encoded Girl images at 0.5 bit / pixel, 1 bit/pixel and 2 bits/pixel. Figure 4.9(a)-(c) show encoded Chemical Plant images at 0.5 bit/pixel, 1 bit/pixel and 2 bits/pixel. As expected, the performance of this model (NC1) is consistent with the conventional transform coding methods.

4.8 Classification Transform Coding

We have mentioned in the last section that in order to increase the transmission efficiency and ease the burden of buffer storage, the test image is divided into small image blocks. Then each image block is transform coded independently. In the classification transform coding, one additional step is added to allocate flexible bits to each transform channel quantizer. It says each image block is classified as belonging to one of K predetermined classes according to its own activity. The activity in the image block is measured by the variance of that image block. For a given class of images the probability distribution function of all the image block variances can be predetermined. Note that this probability distribution function would depend on the size of the image block used in transform coding.

The transform signal in each transform channel is processed as before except that the signal in the transform domain is quantized according to the classification of that image block. In this way, the transform channel quantizer is adapted to the changes of activity in the image blocks. Image blocks with high activity are assigned more quantizer bits than those of low activity. Therefore the quantizer bits are employed more efficiently. This scheme is somewhat analogous to the adaptive classification hybrid coding described in Chapter three.

This scheme also requires the classification information to be transmitted to the receiver. This can be done by sending an extra $\log_2 K$ bits per image block or $\frac{1}{N^2} \log_2 K$ bits/pixel (where K stands for the number of classes and N is the size of an image block). Experimentally, this value turns out to be very small. For example, in the case of K = 4 and M = 16,

the overhead for its transmission is only $\frac{\log_2 4}{16 \times 16} = \frac{1}{128}$ bit / pixel.

Let us define $p_k \triangleq$ probability of the kth class and also assume for each class k, the image model parameters have been predetermined and are known. Then following the same development of adaptive classification hybrid coding, we can write the expected average distortion for each image block as

$$D = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{K} \sigma^2_{v_{i,j}}(k) f(b_{i,j,k}) p_k \tag{4.8-1}$$

where $b_{i,j,k} \geq 0$ and represents the ijth transform channel quantizer bit and kth class.

Another constraint could be the expected average bit rate must be constant, i.e.,

$$\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{K} b_{i,j,k} p_k = P \; . \tag{4.8-2}$$

The above bit allocation problem can be solved along the same lines in the previous section.

Figure 4.10 shows a typical bit allocation pattern for two-class problem with equal probabilities ($p_k = 0.5$) for each class and size of the image block was 64×64. Figure 4.11(a) shows the corresponding 4×4 classification map and image block measure variances. Classification map and image block measure variances for the Chemical Plant image with same kind of strategy is shown in Figure 4.12(b).

The scheme requires the following additional steps compared to ordinary transform coding.

1. Measurement of the variance of each image block.
2. Classification of each image block to one of K classes based on the assumption of p_k.
3. Storing of K bit allocation tables, one for each class.

Results from several simulations show that this scheme gives, relatively, a large improvement in SNR for lower bit rates $\approx$0.5 bit / pixel and relatively smaller image block sizes. At higher bit rates ($\cong$2 bits/pixel) and larger image block sizes, the improvement in SNR becomes less significant if at all, and is evidenced by appearance of a "blocking" effect in the encoded image. This is because as the image block size is increased, the ambiguity in classifying it into different classes increases since the activity in a large image block is no longer uniform. Generally, the following conclusions can be made.

1. Classification transform coding scheme can improve the SNR over the standard transform coding (based on NC1 model) for simple structured image (like Girl image).
2. The complexity of implementing this scheme is marginal. Bit allocation is wholly determined by the model with a proper adjustment of the actual image block variance.
3. As the number of classes goes up, the quality of the reconstructed image improves if the size of image block is much less than the actual size of the test image.

Tables 4.2 and 4.3 list the results of classification transform encoded Girl and Chemical Plant images, both with 64 × 64 image block size and two classes. The rate distortion curves of this scheme are shown in Figures 4.6 and 4.7. Finally, the encoded images at 0.5 bit / pixel, 1 bit/pixel and 2 bits/pixel are shown in Figures 4.8 and 4.9.

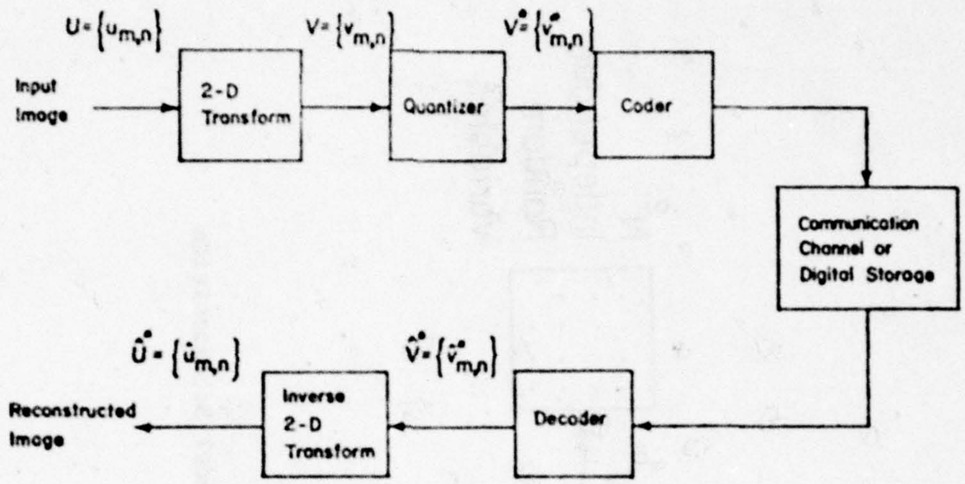


Figure 4.1 Conventional Image Transform Coding System

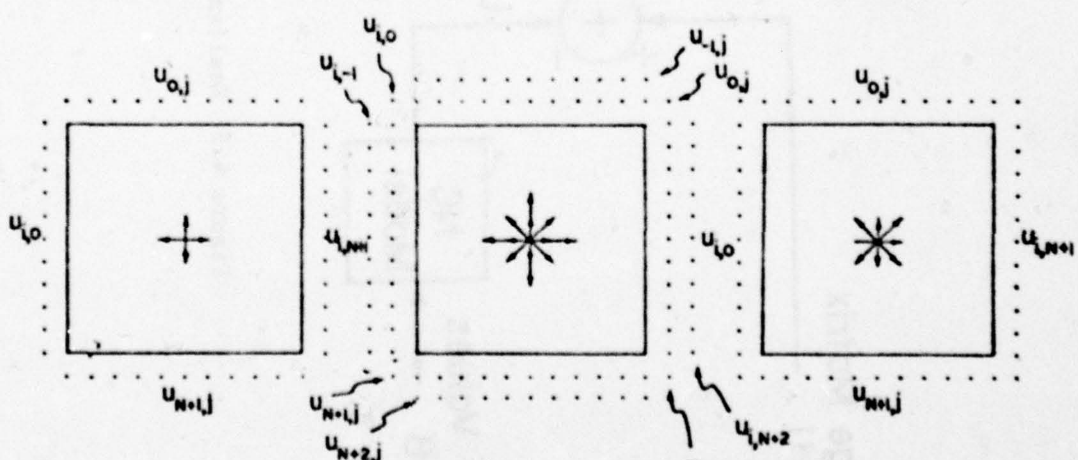


Figure 4.2 Spatial Structures of Three Noncausal Image Representations
(a) NC1 (b) NC2 (c) NC3

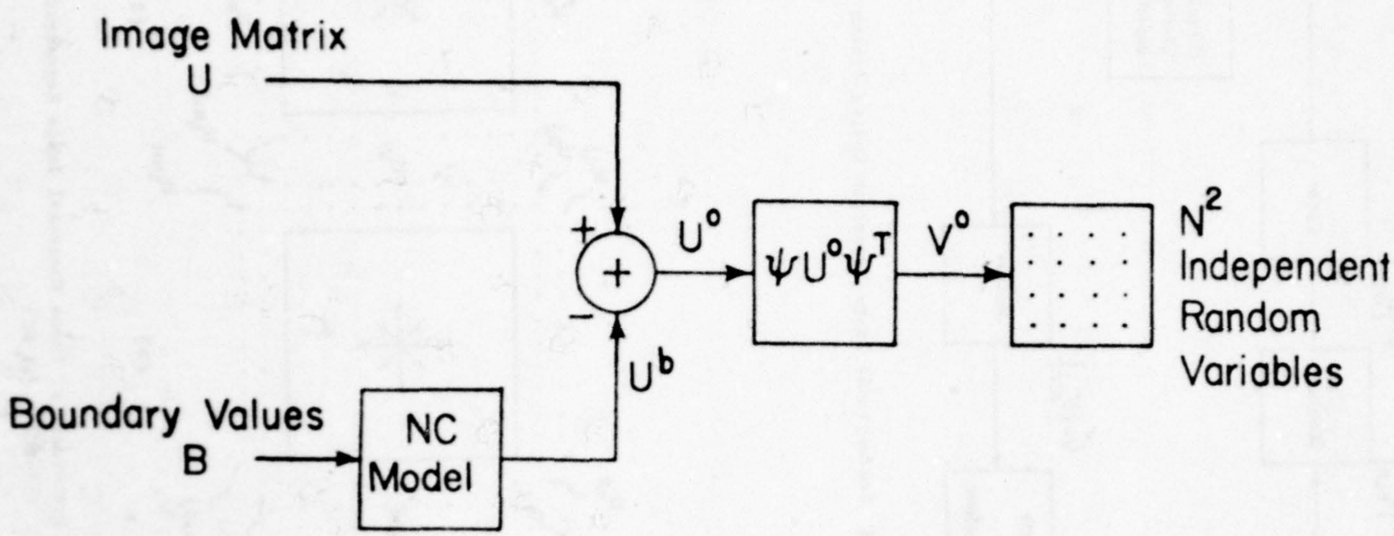


Figure 4.3 Realization of Noncausal Model Decomposition

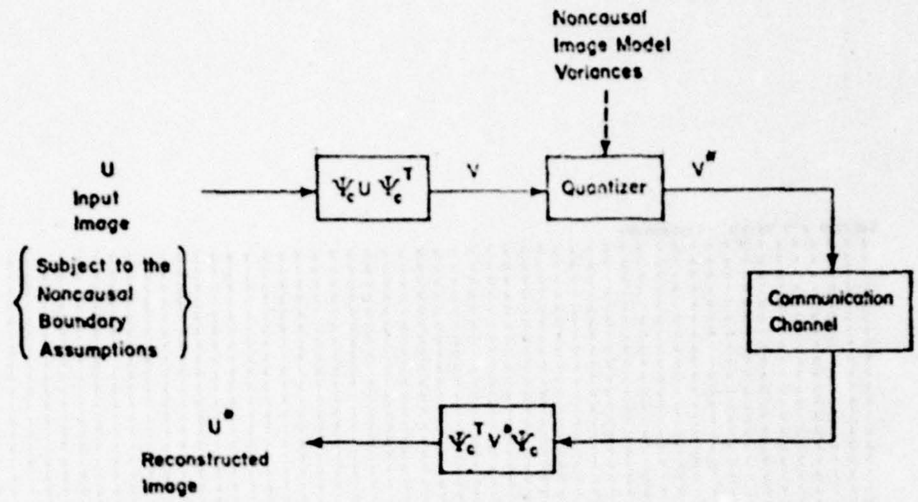


Figure 4.4(a) Cosine Transform Coding via Two-Dimensional Noncausal Model

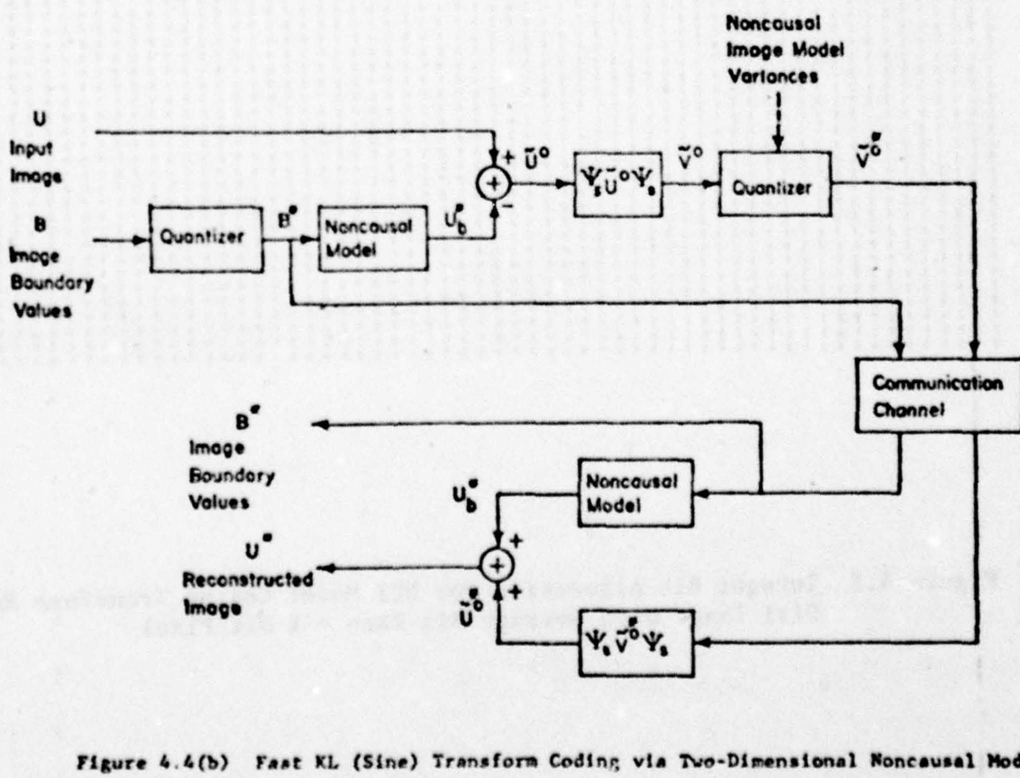


Figure 4.4(b) Fast KL (Sine) Transform Coding via Two-Dimensional Noncausal Model

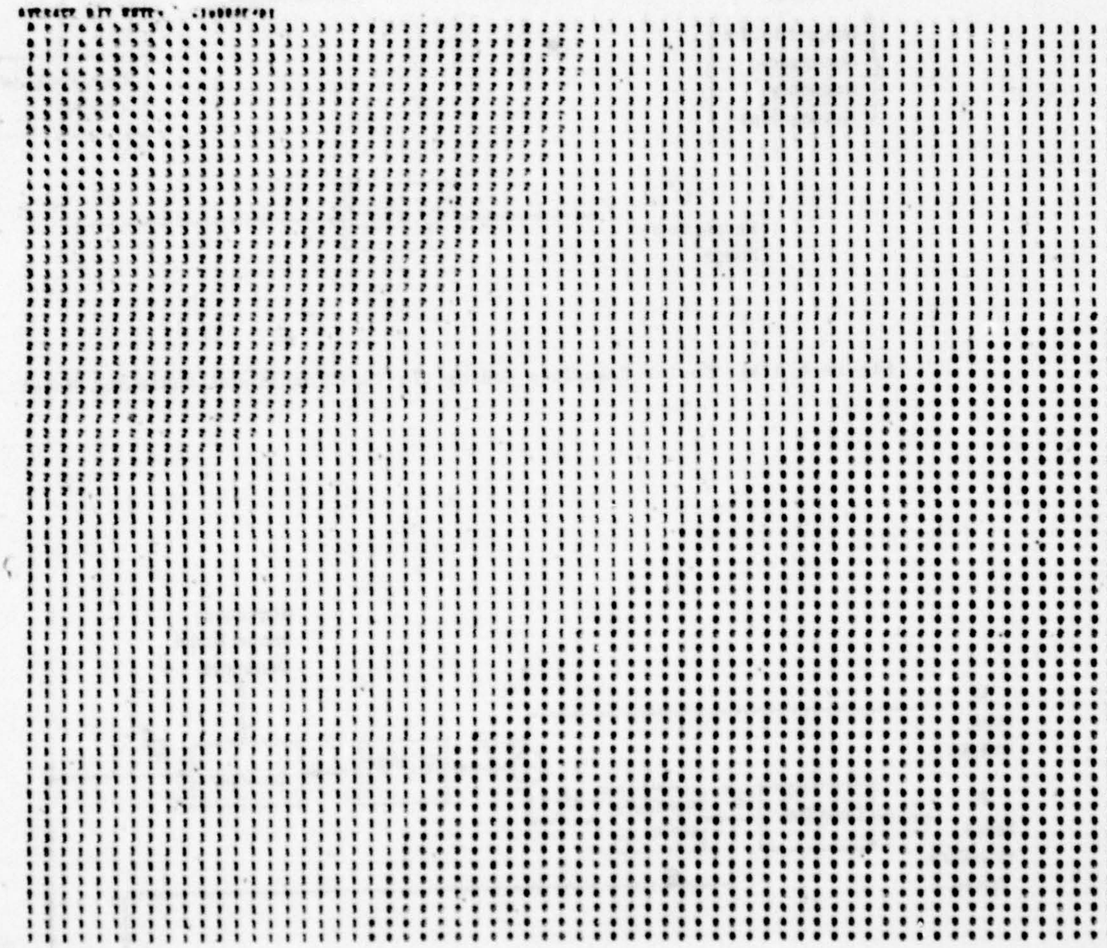


Figure 4.5 Integer Bit Allocation for NC1 Model Cosine Transform Encoded Girl Image with Average Bit Rate = 1 Bit/Pixel

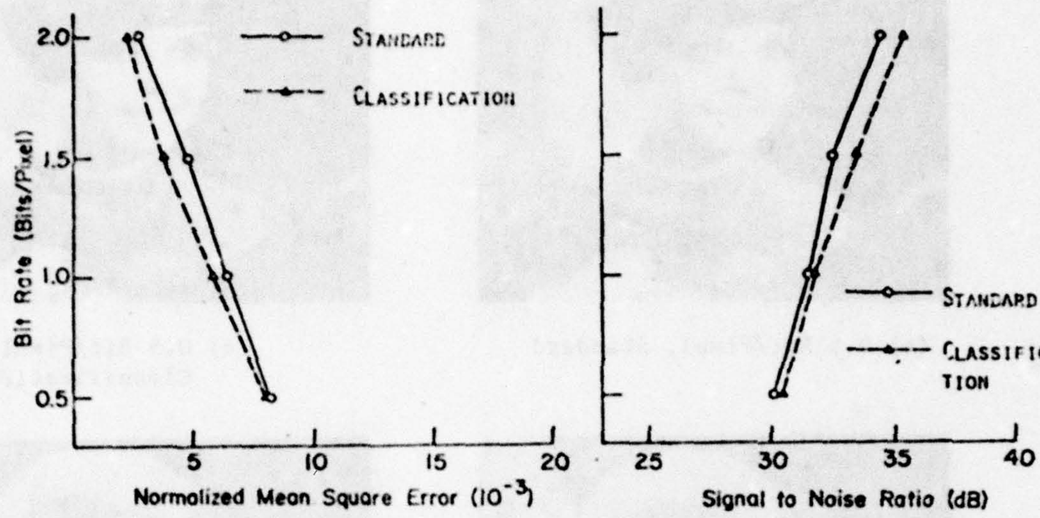


Figure 4.6 Transform Coding Results of *NC1 Model of the Girl Image with Image Block Size* 64 x 64 (a) Rate Distortion Curve (b) Bit Rate versus SNR

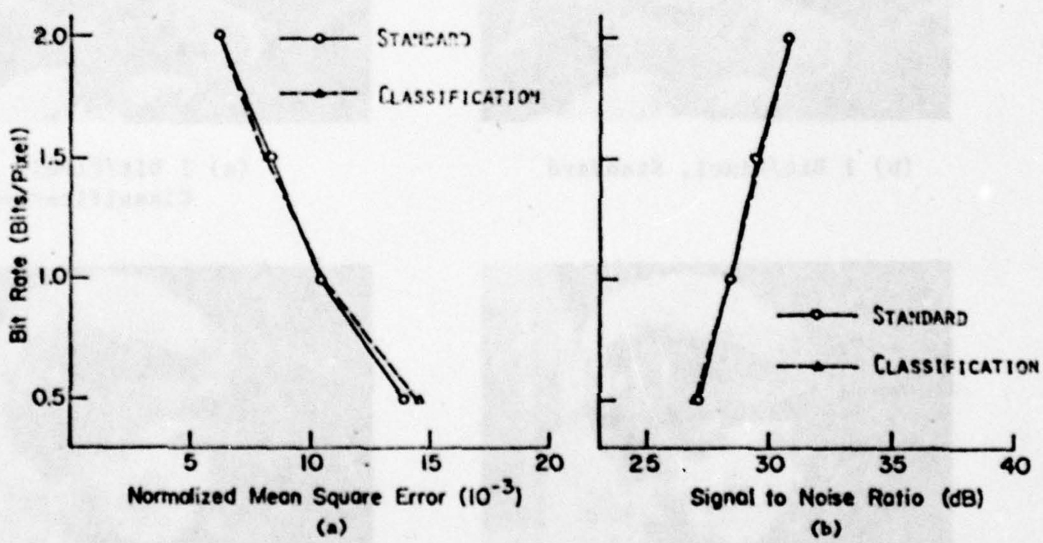


Figure 4.7 Transform Coding Results of NC1 Model of the Chemical Plant Image with Image Block Size 64 x 64 (a) Rate Distortion Curve (b) Bit Rate versus SNR

(a) 0.5 Bit/Pixel, Standard

(d) 0.5 Bit/Pixel, Classification

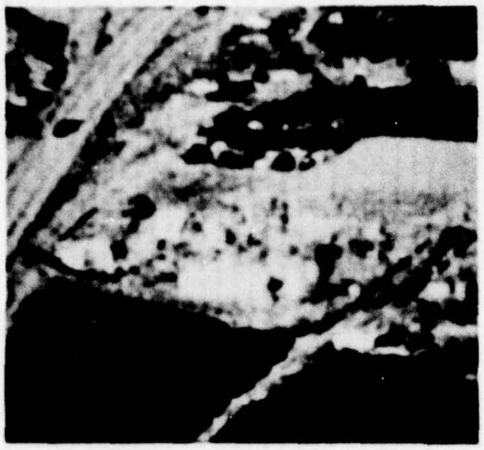
(b) 1 Bit/Pixel, Standard

(e) 1 Bit/Pixel, Classification

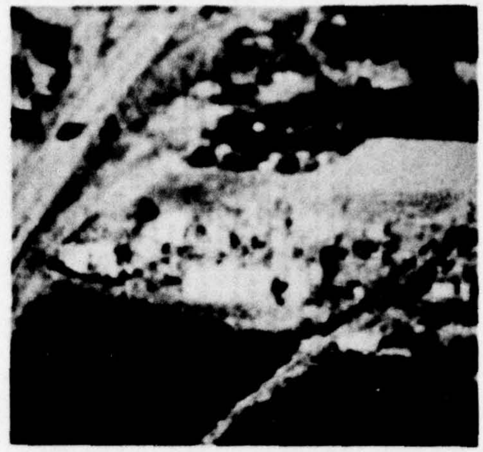
(c) 2 Bits/Pixel, Standard

(f) 2 Bits/Pixel, Classification

Figure 4.8 Cosine Transform Encoded Girl Image



(a) 0.5 Bit/Pixel, Standard



(d) 0.5 Bit/Pixel, Classification

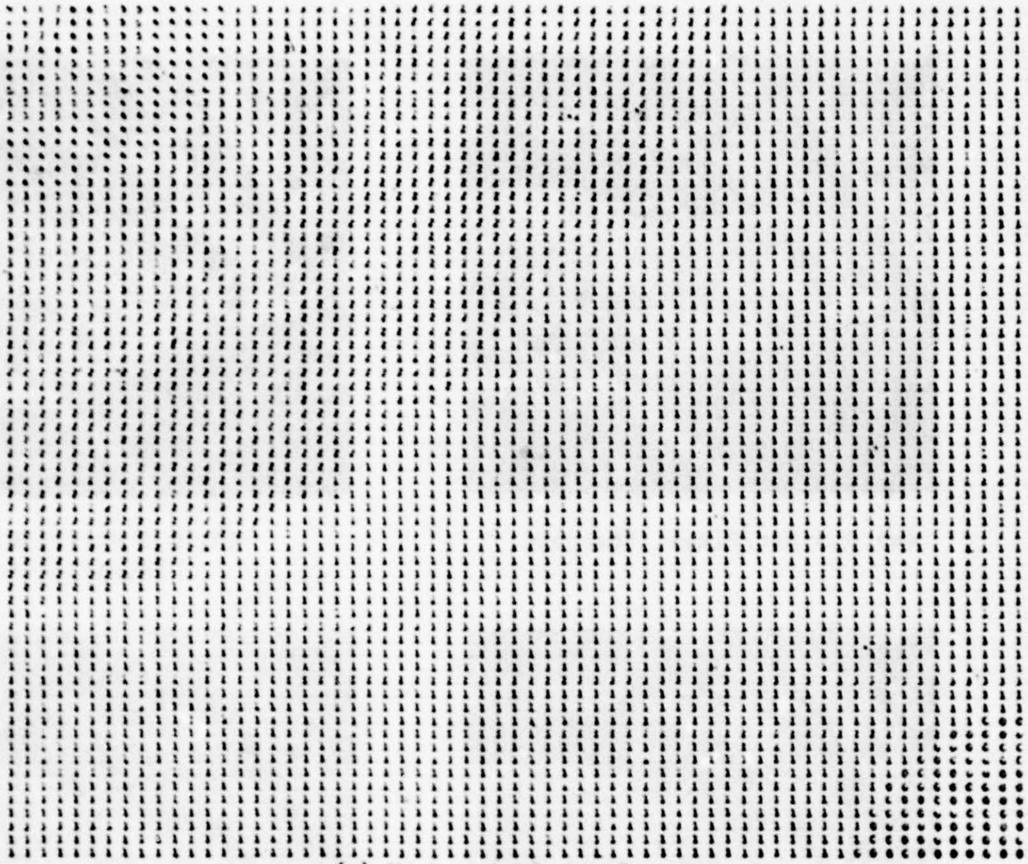
(b) 1 Bit/Pixel, Standard

(e) 1 Bit/Pixel, Classification

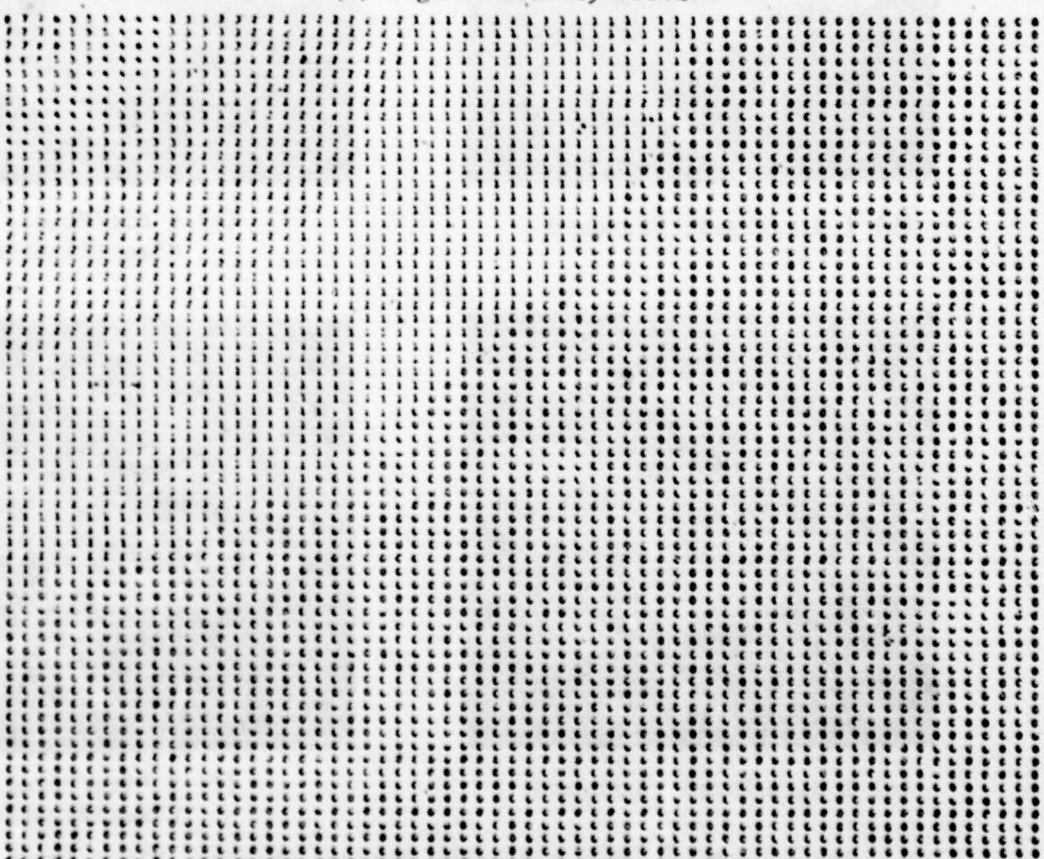
(c) 2 Bits/Pixel, Standard

(f) 2 Bits/Pixel, Classification

Figure 4.9 Cosine Transform Encoded Chemical Plant Image



(a) Higher Activity Portion



(b) Lower Activity Portion

Figure 4.10 Bit Allocation with Classification at Bit Rate = 1 Bit/Pixel (k=2, Girl Image and 64 × 64 Image Block Size)

2	1	1	1	572.26	2166.34	3136.82	1878.98
2	1	2	2	659.34	1639.23	1199.76	929.15
2	1	2	2	1237.50	1649.39	1190.22	1404.61
1	2	1	1	1988.18	1066.08	5779.13	3405.83

(a)

2	2	2	2	1250.41	780.69	646.67	856.33
2	2	2	1	1209.60	984.43	1171.79	1540.41
1	1	1	1	1462.82	2897.02	2053.08	1366.84
1	1	1	2	2596.36	2464.29	1517.38	1051.94

(b)

Figure 4.11 Calssification Map and Image Block Measurement Variances with Two Classes (Equal Probabilities $P_k = 0.5$)
(a) Girl Image (b) Chemical Plant Image.
where 2 indicates lower activity and 1 indicates higher activity, size of the classification map is obtained as $(\frac{256}{N}) \times (\frac{256}{N})$ and $N = 64$ in this example.

Table 4.1 Summary of Important Formulas of Noncausal Represented Image Random Fields

Model	Cosine	Sine
NC1 ARMA Model	$\overline{U} = J^{-1}\overline{\epsilon}$ (4.4-5) $J = (I \otimes Q_c + Q_c \otimes I)$ (4.4-5a) $Q_c = \begin{bmatrix} \tfrac{1}{2}-\alpha & -\alpha & & \\ -\alpha & \tfrac{1}{2} & \ddots & \\ & \ddots & \tfrac{1}{2} & -\alpha \\ & & -\alpha & \tfrac{1}{2}-\alpha \end{bmatrix}$ (4.4-3) $\overline{V} = (\Psi_s \otimes \Psi_c)\overline{U}$ $E[v_{i,j}^2] = \beta^2 \frac{\left(\lambda_{c_{1i}} + \lambda_{c_{1j}}\right)}{\left(\lambda_{c_i} + \lambda_{c_j}\right)^2}$ (4.4-12) $\lambda_{c_i} = \tfrac{1}{2} - 2\alpha \cos \frac{(i-1)\pi}{N}$ (4.4-3a)	$\overline{U} = \overline{U}^o + \overline{U}_b = J^{-1}\overline{\epsilon} + J^{-1}\overline{B}$ (4.3.1-7) $\overline{B}$ (4.3.1-3) $J = I \otimes Q_s + Q_s \otimes I$ (4.3.1-5) $Q_s = \begin{bmatrix} \tfrac{1}{2} & -\alpha & & \\ -\alpha & \ddots & \ddots & \\ & \ddots & \ddots & -\alpha \\ & & -\alpha & \tfrac{1}{2} \end{bmatrix}$ (4.3.1-10) $\overline{V}^o = (\Psi_s \otimes \Psi_s)\overline{U}^o$ (4.3.1-10) $E[v_{ij}^{o2}] = \beta^2 \frac{\left(\lambda_{s_{1i}} + \lambda_{s_{1j}}\right)}{\left(\lambda_{s_i} + \lambda_{s_j}\right)^2}$ (4.3.1-13) $\lambda_{s_i} = \tfrac{1}{2} - 2\alpha \cos \frac{i\pi}{N+1}$ (4.3.1-2)
NC2 ARMA Model	$\overline{U} = J^{-1}\overline{\epsilon}$ $J = I \otimes H + H \otimes I + 2Q_c \otimes Q_c$ $H = Q_c^2 + H^b$ $\overline{V} = (\Psi_c \otimes \Psi_c)\overline{U}$ $E[v_{ij}^2] = \beta^2 \frac{\left(\lambda_{c_{1i}} + \lambda_{c_{1j}}\right)^2}{\left(\lambda_{c_i} + \lambda_{c_j}\right)^4}$ Q_c, λ_{c_i} same as NC1 ARMA Model	$\overline{U} = \overline{U}^o + \overline{U}_b = J^{-1}\overline{\epsilon} + J^{-1}\overline{B}$ (4.3.2-6) $\overline{B}$ (4.3.2-3) $J = I \otimes H + H \otimes I + 2Q_s \otimes Q_s$ (4.3.2-4a) $H = Q_s^2 + H^b$ (4.3.2-8) $\overline{V}^o = (\Psi_s \otimes \Psi_s)\overline{U}^o$ $E[v_{ij}^{o2}] = \beta^2 \frac{\left(\lambda_{s_{1i}} + \lambda_{s_{1j}}\right)^2}{\left(\lambda_{s_i} + \lambda_{s_j}\right)^4}$ (4.3.2-13) Q_s, λ_{s_i} same as NC1 ARMA Model
NC3 Minimum Variance Model	$\overline{U} = J^{-1}\overline{\epsilon}$ $J = Q_c \otimes Q_c$ $Q_c = \begin{bmatrix} 1-\alpha & -\alpha & & \\ -\alpha & 1 & \ddots & \\ & \ddots & 1 & -\alpha \\ & & -\alpha & 1-\alpha \end{bmatrix}$ $\overline{V} = (\Psi_c \otimes \Psi_c)\overline{U}$ $E[u_{ij}^2] = \frac{\beta^2}{\lambda_{c_i}\lambda_{c_j}}$ $\lambda_{c_i} = 1 - 2\alpha \cos \frac{(i-1)\pi}{N}$	$\overline{U} = \overline{U}^o + \overline{U}_b = J^{-1}\overline{\epsilon} + J^{-1}\overline{B}$ (4.3.3-7) $\overline{B}$ (4.3.3-2) $J = Q_s \otimes Q_s$ (4.3.3-6) $Q_s = \begin{bmatrix} 1 & -\alpha & & \\ -\alpha & \ddots & \ddots & \\ & \ddots & \ddots & -\alpha \\ & & -\alpha & 1 \end{bmatrix}$ $\overline{V}^o = (\Psi_s \otimes \Psi_s)\overline{U}^o$ (4.3.3-10) $E[v_{ij}^{o2}] = \frac{\beta^2}{\lambda_{s_i}\lambda_{s_j}}$ (4.3.3-11) $\lambda_{s_i} = 1 - 2\alpha \cos \frac{i\pi}{N+1}$ (4.3.1-2)

Table 4.2 Girl Image, Cosine Transform Coding Results, NCl Model with 64 x 64 Image Block

Bit Rate	Standard		Classification	
	N.M.S.E.(%)	SNR	N.M.S.E.(%)	SNR
0.5	0.8213	30.396	0.8208	30.399
1.0	0.6399	31.480	0.5906	31.828
1.5	0.4929	32.614	0.3956	33.569
2.0	0.3075	34.662	0.2556	35.465

Table 4.3 Chemical Plant Image, Cosine Transform Coding Results, NCl Model with 64 x 64 Image Block

Bit Rate	Standard		Classification	
	N.M.S.E.(%)	SNR	N.M.S.E.(%)	SNR
0.5	1.4067	27.214	1.4711	27.020
1.0	1.0443	28.508	1.0699	28.403
1.5	0.8320	29.495	0.8087	29.619
2.0	0.6106	30.830	0.6244	30.742

CHAPTER FIVE

IMAGE MODELS FOR FEATURE CODING

5.1 Introduction

In this chapter we consider an image coding technique which preserves the attractive characteristics of standard transform coding, i.e., energy packing and channel error immunity, as well as improves the visual appearance of an encoded image. This technique is called Feature transform coding. The basic idea behind this technique lies in segmenting an image which is considered to be a composition of two parts, deterministic and stochastic, where the stochastic part is dealt with by known statistical analysis according to the image model algorithms developed in Chapter 4 and the deterministic part (which could be called features) is obtained as the residual after the stochastic part is removed.

Other two source modeling of images has been considered earlier by Schreiber [62], Yan and Sakrison [74]. The present approach is somewhat more general and is based on stochastic models of images by partial difference equations.

In section 5.2 we demonstrate the concept of feature extraction via image model operators. In section 5.3 we then relate the transform coding scheme with the feature extraction concept and a set of operations performed with this new technique is described in detail. Finally, in section 5.4, we present results of encoded images and compare them with standard NC1 Cosine transform encoded image at the same bit rate.

5.2 Feature Extraction via Image Models

The problem of feature extraction is to determine the rough profile of an object within a given image. It is noted that most important

informations of an image lie on features which contribute the visual appearance of an image. Here a new technique is presented in which the edge points (feature) is detected from the knowledge of image models.

5.2.1 Image Modeling Operator

A given image, U, can be treated as a sample function of a two-dimensional random field. We consider the random field as a composition of two parts, the first part representing the relatively smooth variations within the image which can be described by stochastic image models and the other representing the unpredictable part or deterministic part called the features or edges. Thus, we can write the image U as

$$U = U_s + U_d \tag{5.2.1-1}$$

where U_s and U_d represent the stochastic and deterministic parts respectively.

Typically, the stochastic part representing smooth variations consists of background and low spatial frequency information and the deterministic part describes the abrupt changes (large variations) usually caused by the distinct edges of an object. Therefore, a more effective coding method would be to segment the image into the said parts and code them efficiently.

Typically, an infinite, stationary random field may be represented by its spectral density function (SDF). Several candidates have been proposed in Table 2.1. Let $\mathcal{L}$ denote an operator acting on a discrete image. Denoting ε as the output signal after applying this operator, we can write

$$\varepsilon_s \triangleq \mathcal{L}[U_s], \quad \varepsilon_d \triangleq \mathcal{L}[U_d] . \tag{5.2.1-2}$$

A common example of $\mathcal{L}$ is the Laplacian operator,† which takes the difference of an image element with an algebraic average of its nearest four neighbors. Most feature extraction (or edge detection) problems could be formulated as a segmentation of U_s and U_d in the above equation, of course, with different operators.

Letting t be the arbitrary selected threshold which is employed to classify $\varepsilon(i,j)$, the (i,j)th component of ε, the segmentation criterion is

$$\varepsilon_s(i,j) = \begin{cases} \varepsilon(i,j) & \text{if } |\varepsilon(i,j)| < t \\ t & \text{if } |\varepsilon(i,j)| \geq t \end{cases} \tag{5.2.1-3}$$

$$\varepsilon_d(i,j) = \begin{cases} 0 & \text{if } |\varepsilon(i,j)| < t \\ \varepsilon(i,j) - \text{Sign}\ \varepsilon(i,j) & \text{if } |\varepsilon(i,j)| \geq t \end{cases} \tag{5.2.1-4}$$

Thus we compare $|\varepsilon(i,j)|$ with t, and if $|\varepsilon(i,j)| \geq t$, then the point (i,j) belongs to the set of boundary points; otherwise, it belongs to the stationary part. The values assigned to ε_s and ε_d are according to the above equations.

---

† The Laplacian of a continuous function f(x,y) is defined as

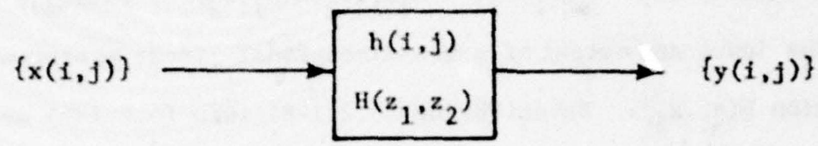
$$\varepsilon(x,y) \triangleq \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

and its discrete equivalent is given by the difference equation

$$\varepsilon_{i,j} = (f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}) - 4f_{i,j}$$

where $f_{i,j} \cong f(x_i,y_j)$, $\varepsilon_{i,j} \cong \varepsilon(x,y)\Delta x\Delta y$, Δx and Δy are the finite difference intervals in the x and y coordinates respectively.

There are two simple ways of selecting the threshold t. First, using the histogram of $|\varepsilon(i,j)|$, a threshold value is picked such that a desired percentage of total image elements would be detected as edge points. Second, we could let $t = c\sigma$, c = constant, σ = standard deviation of $|\varepsilon(i,j)|$. Now, let us consider the selection of $\mathcal{L}$.



Let $\{x(i,j)\}$ and $\{y(i,j)\}$ be the input and output sequences, respectively, of a two-dimensional linear system with transfer function $H(z_1,z_2)$. The SDFs of $\{x(i,j)\}$ and $\{y(i,j)\}$ are related by

$$S_y(z_1,z_2) = H(z_1,z_2)H(z_1^{-1},z_2^{-1})S_x(z_1,z_2) \ . \qquad (5.2.1\text{-}5)$$

For an input white noise field $\{w(i,j)\}$, the output would be a stationary random field, say $\{s(i,j)\}$, such that its SDF is given by

$$S_s(z_1,z_2) = H(z_1,z_2)H(z_1^{-1},z_2^{-1})S_w(z_1,z_2) \ . \qquad (5.2.1\text{-}6)$$

For the sake of simplicity,the SDF of the white noise field is taken to be unity. Then (5.3.1-6) becomes

$$S_s(z_1,z_2) = H(z_1,z_2)H(z_1^{-1},z_2^{-1}) \ . \qquad (5.2.1\text{-}7)$$

Therefore the transfer function can be viewed as

$$H(z_1,z_2) = [S_s(z_1,z_2)]^{\frac{1}{2}} \ . \qquad (5.2.1\text{-}8)$$

Taking a two-dimensional Z transform on both sides of (5.2.1-1) yields

$$V(z_1,z_2) = V_s(z_1,z_2) + V_d(z_1,z_2) \ , \tag{5.2.1-9}$$

where $$V \triangleq Z\{U\},\ V_s \triangleq Z\{U_s\} \text{ and } V_d \triangleq Z\{U_d\}. \tag{5.2.1-9a}$$

It is obvious that $V_s(z_1,z_2) = H(z_1,z_2)W(z_1,z_2)$ if $\{w(i,j)\}$ and $\{u_s(i,j)\}$ are the input and output of a two-dimensional linear system with transfer function $H(z_1,z_2)$. Substituting (5.2.1-8) into (5.2.1-9) we get

$$V(z_1,z_2) = [S_s(z_1,z_2)]^{\frac{1}{2}} W(z_1,z_2) + V_d(z_1,z_2) \tag{5.2.1-10}$$

where $W(z_1,z_2)$ is the two-dimensional Z transform of the white noise field.

Rearranging the terms of (5.2.1-10), we have

$$[S_s(z_1,z_2)]^{-\frac{1}{2}}V(z_1,z_2) = W(z_1,z_2) + [S_s(z_1,z_2)]^{-\frac{1}{2}}V_d(z_1,z_2) \tag{5.2.1-11}$$

Denoting $\epsilon(z_1,z_2) \triangleq [S_s(z_1,z_2)]^{-\frac{1}{2}}V(z_1,z_2)$ and taking inverse two-dimensional Z transform on both sides of (5.2.1-11), we get

$$\varepsilon(i,j) = \varepsilon_s(i,j) + \varepsilon_d(i,j) \quad \forall\ i,j \tag{5.2.1-12}$$

where ε_d is the inverse Z transform of $S_s^{-\frac{1}{2}}V_d$, and ε_s is the white noise field. For finite image, we define a two-dimensional transform of the image as

$$V \triangleq \overline{\Psi}[U] = \Psi U \Psi^T \tag{5.2.1-13}$$

where superscript T indicates conjugate and transpose operations and $\overline{\Psi}$ is the unitary KL transform of U. (The above equation implies $\overline{\Psi}$ is a

separable transform, which will be true for our image models). From the foregoing discussion, if we define

$$\mathcal{L} = \Psi^{-1} S_s^{-\frac{1}{2}} \Psi \qquad (5.2.1\text{-}14)$$

where superscript -1 indicates inverse operation and S_s is the SDF of stationary random field, then $\mathcal{L}$ would be the desired edge detection operator.

5.2.2 Approximate Rational Function Operator

The SDF of NC1 image model is given by

$$S_s(z_1,z_2) = \frac{1 - r\alpha(z_1 + z_1^{-1} + z_2 + z_2^{-1})}{[1 - \alpha(z_1 + z_1^{-1} + z_2 + z_2^{-1})]^2} \qquad (5.2.2\text{-}1)$$

where $r\alpha = \alpha_1$ and $\beta^2 = 1$, to be consistent with the formula given in Table 2.1. If we assume $R(z_1,z_2) = [S_s(z_1,z_2)]^{-\frac{1}{2}}$ then the alternative expression for $R(z_1,z_2)$ is given by (see Appendix B)

$$R(z_1,z_2) = 1 - \sum_{n=1}^{\infty} \frac{1\cdot 3\cdot 5 \cdots (2n-3)}{2^{n-1}(n-1)!}\left(1 - r + \frac{r}{2n}\right) r^{n-1} x^n \,. \qquad (5.2.2\text{-}2)$$

By picking different values of n, we obtain various orders of $R(z_1,z_2)$ with which the edge detection operator could be defined as

$$\mathcal{L} = R^{(n)}(z_1,z_2) \qquad n = 1,2,\ldots \qquad (5.2.2\text{-}3)$$

This implies $\mathcal{L}$ is a two-dimensional nth order difference operator.

5.2.3 Edge Information Subtraction Procedure

We have stated the concepts of image modeling operator and edge points segmentation criterion in the preceding sections. The sequence

of operations needed to be performed may now be listed as follows:

1. Take discrete unitary transform of the input image, $V = \Psi U \Psi^T$, where Ψ denotes the chosen discrete unitary transform.

2. Based on the image model, find the proper representation for the SDF, S_s, in the Ψ domain.

3. Calculate $S_s^{-\frac{1}{2}}$ and multiply it with V, yielding $\mathcal{E} = S_s^{-\frac{1}{2}} V$.

4. Take inverse discrete unitary transform of $\mathcal{E}$, obtaining $\varepsilon = \Psi^{-1} \mathcal{E} \Psi^{-T}$.

5. Pick a reasonable threshold to segment ε_d from ε_s according to equations (5.2.1-3) and (5.2.1-4).

6. Generate edge address map $\mathcal{I}_d$ which is the set of (i,j) coordinates of each detected edge point.

Once the edge information is known, the stationary field U_s and deterministic field U_d of the given image can be obtained by the following fairly obvious way:

7. Take discrete unitary transform of ε_d, getting $\mathcal{E}_d = \Psi \varepsilon_d \Psi^T$.

8. Premultiply $S_s^{\frac{1}{2}}$ with $\mathcal{E}_d$, and define $V_d = S_s^{\frac{1}{2}} \mathcal{E}_d$.

9. Take inverse discrete unitary transform of V_d to obtain $U_d = \Psi^T V_d \Psi$.

10. Subtracting U_d from the input image U, gives the stationary field as $U_s = U - U_d$.

We have considered two transforms, i.e., discrete Fourier transform and discrete Cosine transform as the two-dimensional unitary transform operators in our preliminary feature extraction study. The experiments were performed on the 256 × 256 Girl image in 64 × 64 image blocks. For

comparison, the thresholds for different methods of edge detection were chosen to yield approximately equal number of detected edge points. The formulas for evaluating the value $S_s(k,\ell)$ will be different according to which transform is actually implemented and the image model used. The NC1 model was chosen for all experiments. First we let

$S_s(k,\ell) = S_s\left(z_1 = e^{\frac{-2\pi k}{N}}, z_2 = e^{\frac{-2\pi\ell}{N}}\right)$ be the approximation of the SDF in the discrete Fourier transform domain when N is sufficiently large. However, in the case of Cosine transform, the numerical value of $S_s(k,\ell)$ is taken from the cosine transform variance formula (see Table 4.1). Figures 5.1(a) and 5.1(b) show the resulting images of these two experiments. A significant visual difference can be observed from these two figures. The block effect obtained with the discrete Fourier transform approximation is not present in using the Cosine transform. Hence, it is preferable to use the Cosine transform in the study of feature extraction concepts.

If the approximate rational operator $R^{(n)}$ is used, rather than $S_s^{-\frac{1}{2}}$, $\varepsilon(i,j)$ could be calculated directly in the spatial domain (because $R^{(n)}$ becomes a two-dimensional nth order difference operator). Segmentation of $\varepsilon(i,j)$ could be performed as before. The spatial structure of $R^{(n)}$ as a difference operator is shown for various values of n in Appendix B.

5.2.4 Image Enhancement

A typical image is usually made up of objects having edges at which the brightness changes abruptly. It has been known to investigators in the image processing field that the human visual system is sensitive to these edges. By this argument a "good" image should result if these

edge points can be isolated and emphasized so as to make the image appear sharp. We have initiated a study towards this end. Following the concept developed in the previous sections, the stationary (stochastic) field and deterministic (feature) field of any given image can be easily generated. The way to "sharpen" the image can be done by the following equation

$$U' = \alpha U_s + \beta U_d \quad \forall\ i,j \qquad (5.2.4\text{-}1)$$

where U_s represents the stationary field which contains mostly background information, and U_d represents the deterministic field which results from the edge information.

Using (5.2.4-1), one can change various values of α and β and achieve different enhancement (sharp or blur) images U'. Figures 5.3II and 5.4II demonstrate the results. The thresholds used in these experiments are given in Table 5.1. It is seen that $U_s + 1.5U_d$ gives the best enhancement image among the six combinations in our study.

The overall block diagram of this system is shown in Fig. 5.2, which consists of two functions, decomposition and synthesis. The stochastic modeling operator $\mathcal{L}$ is determined by a chosen image model, and (5.2.4-1) is implemented by adjusting the two values α and β.

5.3 Transform Coding Using Nonstationary Image Representation

In the foregoing presentation, we considered a new concept of image decomposition, an image can be separated into its stochastic (stationary) and deterministic (feature) components, i.e.,

$$U = U_s + U_d \qquad (5.2.1\text{-}1)$$

One of the possible applications of (5.2.1-1) is to extend it to image coding or data compression. In comparison with current existing coding schemes, including the one presented in Chapter 4, this scheme is found to give better performance in terms of resolution for the same bit rate, and gives more pleasing encoded images to the human viewer.

Let us give an overview of this coding scheme and then describe the various operations in detail. The block diagram of this scheme is given in Figure 5.5. First, U is transformed by a two-dimensional unitary transform. There is a tradeoff between a large image block size to extract sufficient edge information and a small size to save memory and computational effort. We find that a good compromise is achieved by 64 × 64 blocks. Next, applying operator $\mathcal{L}'$ on V (see Fig. 5.5) where V is the transformed image. The output of this operation, ε, is segmented next according to (5.2.1-3) and (5.2.1-4). If $|\varepsilon(i,j)| \geq t$, the location (i,j) is recorded as one of the edge addresses, otherwise it is rejected. The value of ε at the detected edge point is quantized before transmission to the receiver. This quantized version of ε, ε^*, is also needed to calculate V_d^* as $\mathcal{L}^{-1}\varepsilon^*$, where V_d^* is viewed as the deterministic part of the transformed image. Subtracting V_d^* from V, we have V_s, the stochastic part of the transformed image. A conventional transform coding scheme is then implemented on V_s. Finally, V_s^*, ε_d^* together with edge address map, $\mathcal{J}_d$, are used for coding and transmission over the communication channel.

At the receiver, the process proceeds in a fairly obvious way. First, we reconstruct V_d^* based on the received ε_d^* and $\mathcal{J}_d$. Then summing it with V_s^*, yields V^*. Finally, the inverse unitary transform is operated on V^* to recover the input image.

We now describe details of some of the operations that need to be performed.

1) Run Length Coding

The purpose of this operation is to efficiently transmit the edge address map. The entropy of the run length serves as a measure of bit rate required to recover the original edge address map without distortion. For higher threshold, t, fewer pixels are detected as edge points. In this situation, in order to prevent excessively long runs, pseudo runs are inserted if run length exceeds a fixed number (64 in our experiment).

2) Fixed Level Quantization

Essentially we have only one quantizer at this stage and $\varepsilon_d(i,j)$ is assumed to have a Laplacian distribution with mean zero and variance $\sigma^2_{\varepsilon_d}$ which has to be calculated for each value of t, since the values of $\varepsilon_d(i,j)$ would vary with t. Due to the fact that the values of $\varepsilon_d(i,j)$ are usually small, it has been found that a two bit quantizer gives a very accurate reproduction of $\varepsilon_d(i,j)$. We have studied the effect of assigning different bits to the quantizer. Notice that if the number of detected edge points increases, then assigning too many bits to the quantizer would mean that we are not able to allocate sufficient bits to the transform domain quantizers, and this would cause the objectionable low spatial frequency quantization errors in the encoded image.

3) Transform Domain Quantization

In order to accomplish the task of data compression, we must allocate unequal number of bits to the quantizers for each element of V_s. The

fundamental idea of transform coding is that high energy (activity) samples are quantized more accurately than low energy samples. The variances of each element of V_s, the stochastic part of the transformed image, can be estimated from the SDF corresponding to the transform used. The quantization scheme used here is the Lloyd-Max quantizer with the assumption that each element of V_s is Gaussian in distribution.

4) Function of $\mathcal{Z}'$

Referring to Fig. 5.5, it is seen that

$$\varepsilon = \mathcal{Z}'V = \overline{\psi}^{-1} S_s^{-\frac{1}{2}} V . \tag{5.3-1}$$

From the above equation, clearly, we have defined $\mathcal{Z}'$ as

$$\mathcal{Z}' = \overline{\psi}^{-1} S_s^{-\frac{1}{2}} . \tag{5.3-2}$$

Comparing (5.3-2) with (5.2.1-4), it can be seen that $\mathcal{Z}'$ is slightly different from $\mathcal{Z}$, the way presented in Fig. 5.5 is easily realizable.

5) Considerations in Deleting Isolated Point

Edge detection is often affected by noise contained in the original image. If we examine the edge picture (picture composed of the detected edge points only) closely, see Fig. 5.1, we would find many isolated spots spread over the scene. Actually, we have no sharp distinction between a "true edge point" or a "noise" point. A simple edge detection scheme in the presence of small amount of noise is to first implement the edge extraction algorithm, then followed by an algorithm that removes the isolated edges (noise) from the scene. In this way, one could get a much

clearer edge profile. We have included this scheme inside edge detection logic box of Figure 5.5.

An advantage of this transform coding method is that it gives us a flexible way to meet any requirement of getting a good visual image subject to a fixed bit rate. There are three adjustable parameters associated with the entire process, viz,

a. Edge point selection threshold

The basic formula is given by $t = c\sigma$, where σ is the rms value of $\varepsilon(i,j)$, and c is an adjustable constant. When $c \to \infty$ no pixel will be detected as an edge point. Some of the operations in Figure 5.5 could be bypassed in this case and a conventional transform coding method is directly performed on the image U.

b. Edge quantizer's bit (n_e bits/edge)

We have made experiments concerning this fixed integer bit's effect on a bit allocation strategy. The various bit rates used are 1, 2, 3 and 4. The results will be presented in the next section. Experimentally, it has been found that 2 bits (4 levels) gives a proper bit selection.

c. Desired bit rate (n)

It is the desired overall average bit rate in bits/pixel in the final encoded image.

The above mentioned parameters are related by

$$n = n_1 + n_2 + n_3 \qquad (5.3\text{-}3)$$

n_1 : Average bit rate for transform coding of V_s^*

n_2 : Average bit rate for coding the amplitudes of detected edge points

$$= \frac{n_e \times (\text{Number of Detected Edge Points})}{(\text{Number of Image Pixels})} \qquad (5.3\text{-}4)$$

n_3 : Bit rate required to recover the edge address map, $\mathcal{J}_d$.

$$\geq \frac{(\text{Entropy of } \mathcal{J}_d) \times (\text{Number of Detected Edge Points})}{(\text{Number of Image Pixels})} \qquad (5.3\text{-}5)$$

Once the edge point selection threshold is given, the number of detected edge points is automatically determined. For a fixed n_e, n_2 and n_3 can be calculated via (5.3-4) and (5.3-5), respectively. If we desire to have bit rate of n (or $\frac{8}{n}$ data compression) in the final encoded image, n_1 has to be calculated via (5.3-3).

5.4 Experimental Results and Comparisons

A series of computer simulations has been conducted to evaluate the performance of this scheme called Feature transform coding. A 256 × 256 image was first divided into 16 equal-sized image blocks, each of size 64 × 64. Each image block was feature transform coded independently. In all the experiments, Cosine transform based NC1 model was used and several definitions of performance measurements, i.e., SNR, NMSE, etc., are defined in section 3.5 of Chapter 3. Two images of different category (statistics) were used in order to test how sensitive the scheme was to the image content.

The quantities n_e and t are the major parameters in the design of Feature transform coding. In our study, it is done experimentally as presented in the next section.

5.4.1 Feature Transform Coding

In order to find the proper choice of n_e (edge quantizer's bit) for different values of t (threshold), we plot the curves of SNR and NMSE versus various settings of n_e for $t = 1.5\sigma$ and $t = 2\sigma$. Figures 5.6-5.7 and Figures 5.10-5.11 are the results for Girl and Chemical Plant images, respectively. It can be seen from these figures that n_e must be equal to 2, since it gives the minimum NMSE or maximum SNR of all the bit rates. After determining n_e, we also plot the curves of SNR and NMSE versus various thresholds. Figure 5.8 demonstrates the curve for the Girl image, and Figure 5.12 is for the Chemical Plant image. It is observed that the SNR increases, while the threshold decreases for moderate bit rate ($1 \sim 1.5$) of both images. For high bit rate (≈ 2) this phenomenon is still observed for Chemical Plant image, but not for Girl image. This could be because the Chemical Plant image is a complicated scene in which more edge points must be separated from the background and encoded adequately. For low bit rate (≈ 0.5), as expected, this scheme performs slightly better than the standard transform coding (in our case NC1 Cosine transform coding) for higher thresholds only. Based on these observations, it seems that $t = 1.5\sigma$ and $n_e = 2$ serve as the best parameters to be used to encode any given image at rate greater than 0.5 bit / pixel.

The comparisons of Feature transform encoded and standard transform encoded images are shown in Figure 5.14 for the Girl image and corresponding ones for the Chemical Plant image are shown in Figure 5.15. In all the images the Feature transform coder produces a "better" appearing image (at bit rate > 0.5) than the standard transform coder. It is easily seen that there is a significant difference around the Girl's face and at the sides of the freeway in the Chemical Plant image. In general, the edges

are more clearly defined on the Feature transform encoded image. The above discussion is also supported by referring to the coding results shown in Tables 5.2 through 5.9, which give quantitative comparisons of the above two mentioned schemes. For easy reference, the performances of the standard transform coder at different bit rates are listed along with the Feature transform coder. The performance improves by about 1.5 dB at 1 bit rate for both images and by 2dB at 2 bit rate for Chemical Plant only. It is of interest to note that at 2 bit rate Figure 5.14(f) has less SNR value than Figure 5.14(c). However, by directly looking at the pictures, one can say that Figure 5.14(f) gives a better appearing image since the edges are more sharp regardless of the contrast of the background.

5.4.2 Classification Feature Transform Coding

In this method the edge information map $\mathcal{J}_d$ and edge point ε_d are encoded and transmitted as before, except that the stochastic portion of the transformed image, V_s, is classified as belonging to one of K pre-determined classes. In our case, K = 2 and each class is assumed to be equiprobable. The classification maps for the Girl and Chemical Plant images are shown in Fig. 4.11(a) and Fig. 4.11(b). The block diagram of this scheme is analogous to Figure 5.5. However, the Transform Domain Quantization box now contains two types of bit allocation maps, one for each class.

This method gives an additional improvement of SNR over the Feature transform coder about 1 dB at bit rate $\geq$ 1.5 for the Girl image. But it failed for the Chemical Plant image. Figures 5.9 and 5.13 show the coding results versus various thresholds with n_e = 2 assigned to the edge point quantizer. The reason for the discrepancy could be that the Chemical Plant

image is a complicated scene, and once enough detected edge point information has been subtracted from the original nonstationary image field, the resultant image field, i.e. V_s, would tend to have equal variances for each image block, i.e., the corresponding pixels of the remaining field V_s of each image block have approximately equal variances. However, this kind of property is violated for a simple structure image, i.e., Girl image, where after the edge point information is subtracted, some refined work, for example classification, could be applied on V_s if the ultimate goal of coding scheme is to get higher SNR in the final encoded image. We have shown the encoded images of this scheme along with Feature transform encoded images in Figures 5.16 and 5.17.

5.4.3 Conclusions

Based on the above experiments and the theory discussed in Chapter 4, the following conclusions are made:

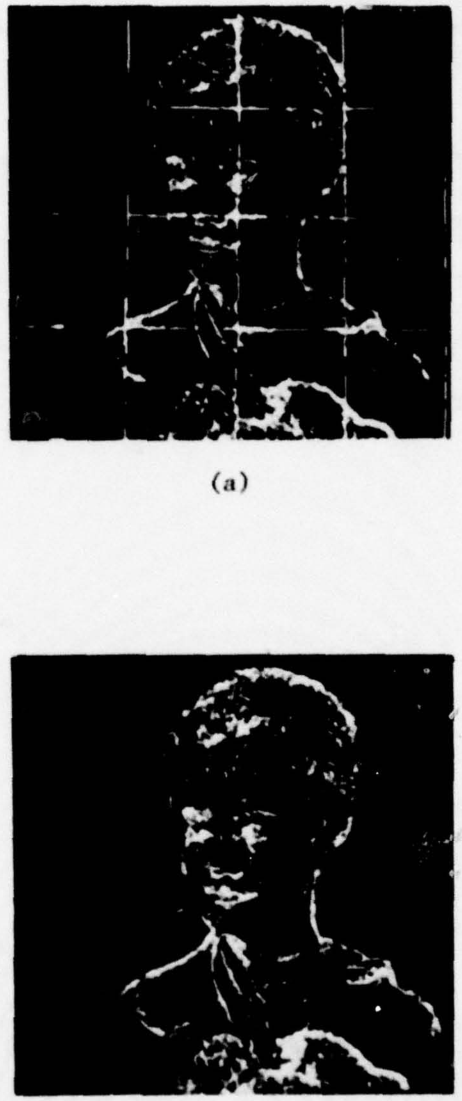
(1) Regardless of the computational complexity, Feature transform coding scheme can improve the signal to noise ratio over the standard transform coding scheme (based on NC1 Cosine transform model) by 1 to 2 dB at bit rates above 1 bit/pixel for usual images.

(2) The complexity of Feature transform coding scheme is simpler than the technique proposed by Yen et al [74] where they ignored the problem of computation time, complexity, and storage. Meanwhile, they had more image dependent parameters that need to be identified. In our case, only two parameters, i.e., n_e and t, are sufficient.

(3) The Feature transform coding scheme is useful when it is desired to maintain the attractive characteristics of the transform coding

scheme, i.e., less sensitivity to the image statistics (scene to scene variation), as well as to incorporate the physiological reaction of the human visual system in the overall communication system design.

(4) The Classification Feature transform coding scheme is suited for simple structured images at bit rate $\geq$ 1 bit/pixel. The complexity over the Feature transform code is marginal and its performance is improved by reducing the size of the image block or, in other words, the number of image blocks is increased.



(a)

(b)

Figure 5.1 Edge Pictures with $t = 2\sigma$ (a) Discrete Fourier Transform (b) Discrete Cosine Transform, where σ is the rms Value of $\varepsilon(i,j)$, and the Total Detected Edge Points are 3612 and 3587 for (a) and (b), respectively.

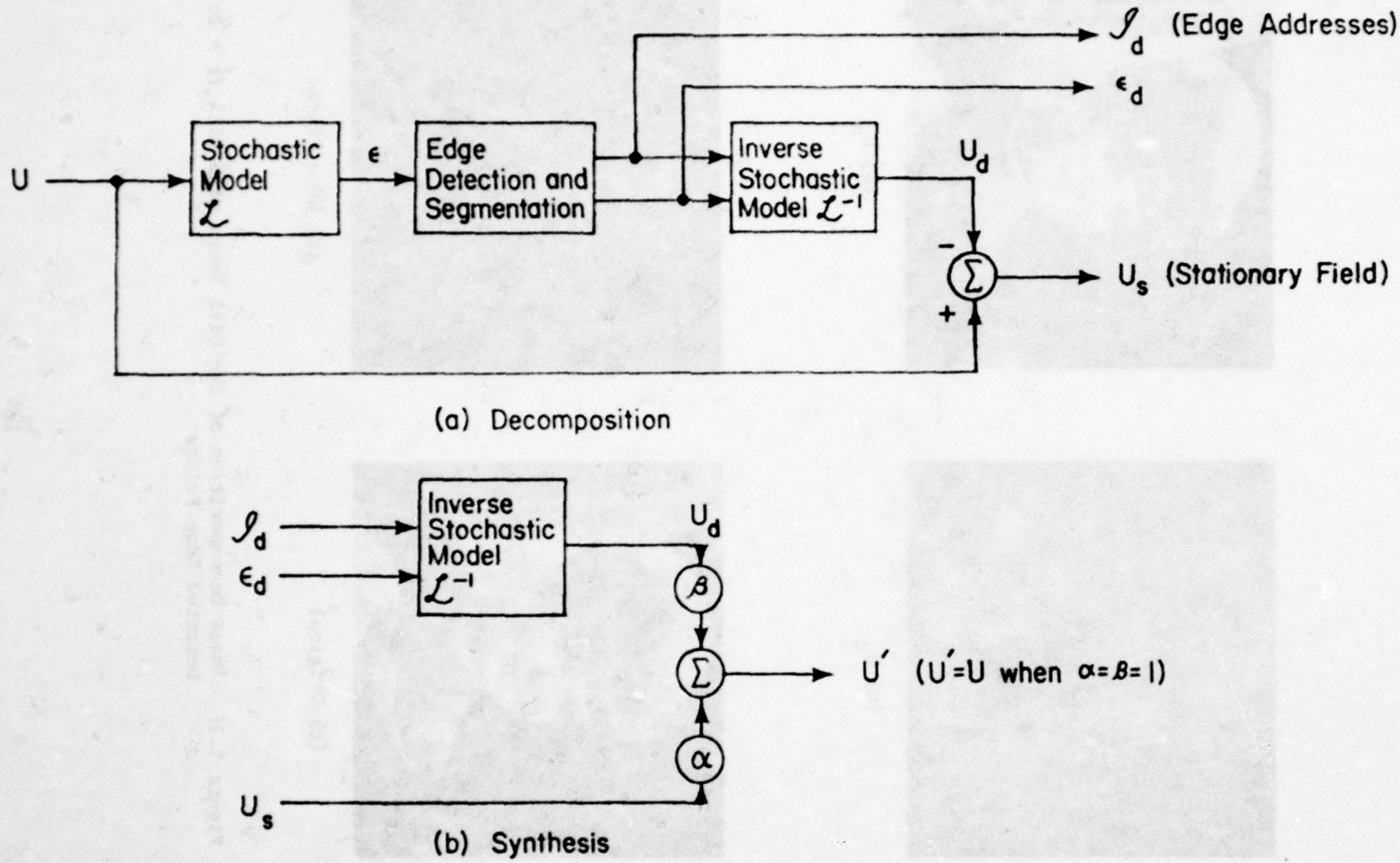


Figure 5.2 Realization of Stochastic Decomposition

(a) U_d

(b) U_s

(c) Original

(d) Edge Pixels

Figure 5.31 Image Decomposition of the Girl Image with 3111 ($t = 2\sigma$) Detected Edge Points

(a) $0.5U_s + 1.5U_d$

(b) $0.75U_s + 1.25U_d$

(c) $U_s + 1.5U_d$

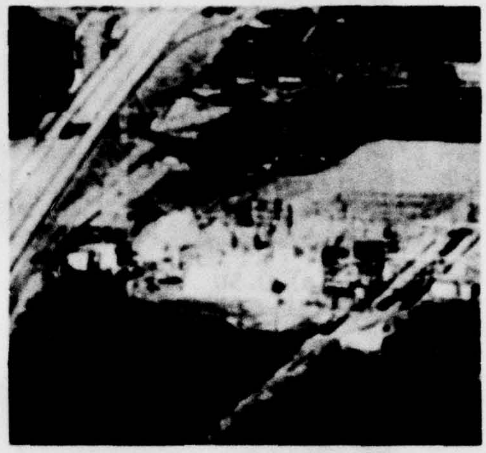
(d) $U_s + 2U_d$

(e) $U_s + 2.5U_d$

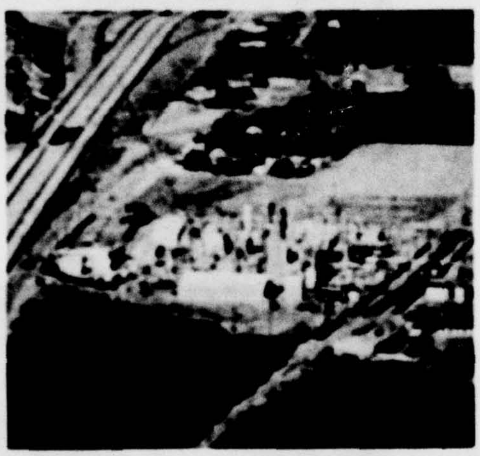
(f) $U_s + 3U_d$

Figure 5.3II Image Enhancement of the Girl Image with 3111 ($t = 2\sigma$) Detected Edge Points

(a) U_d



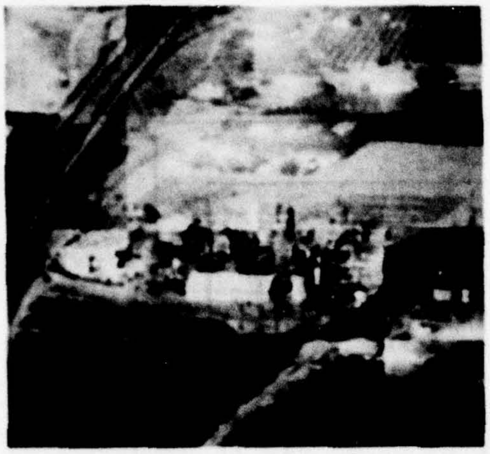
(b) U_s



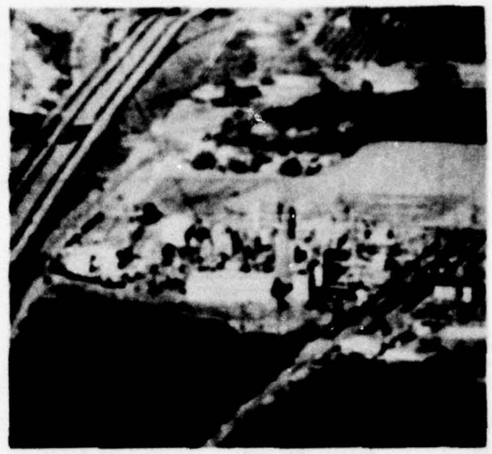
(c) Original

(d) Edge Pixels

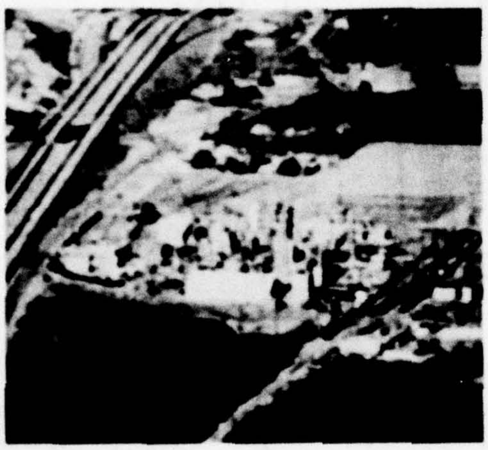
Figure 5.41 Image Decomposition of the Chemical Plant Image with 3653 ($t = 2\sigma$) Detected Edge Points



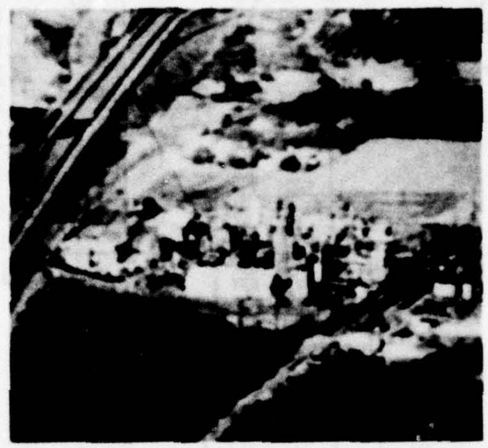
(a) $0.5U_s + 1.5U_d$



(b) $0.75U_s + 1.25U_d$



(c) $U_s + 1.5U_d$



(d) $U_s + 2U_d$

(e) $U_s + 2.5U_d$

$U_s + 3U_d$

Figure 5.4II Image Enhancement of the Chemical Plant Image with 3653 ($t = 2\sigma$) Detected Edge Points

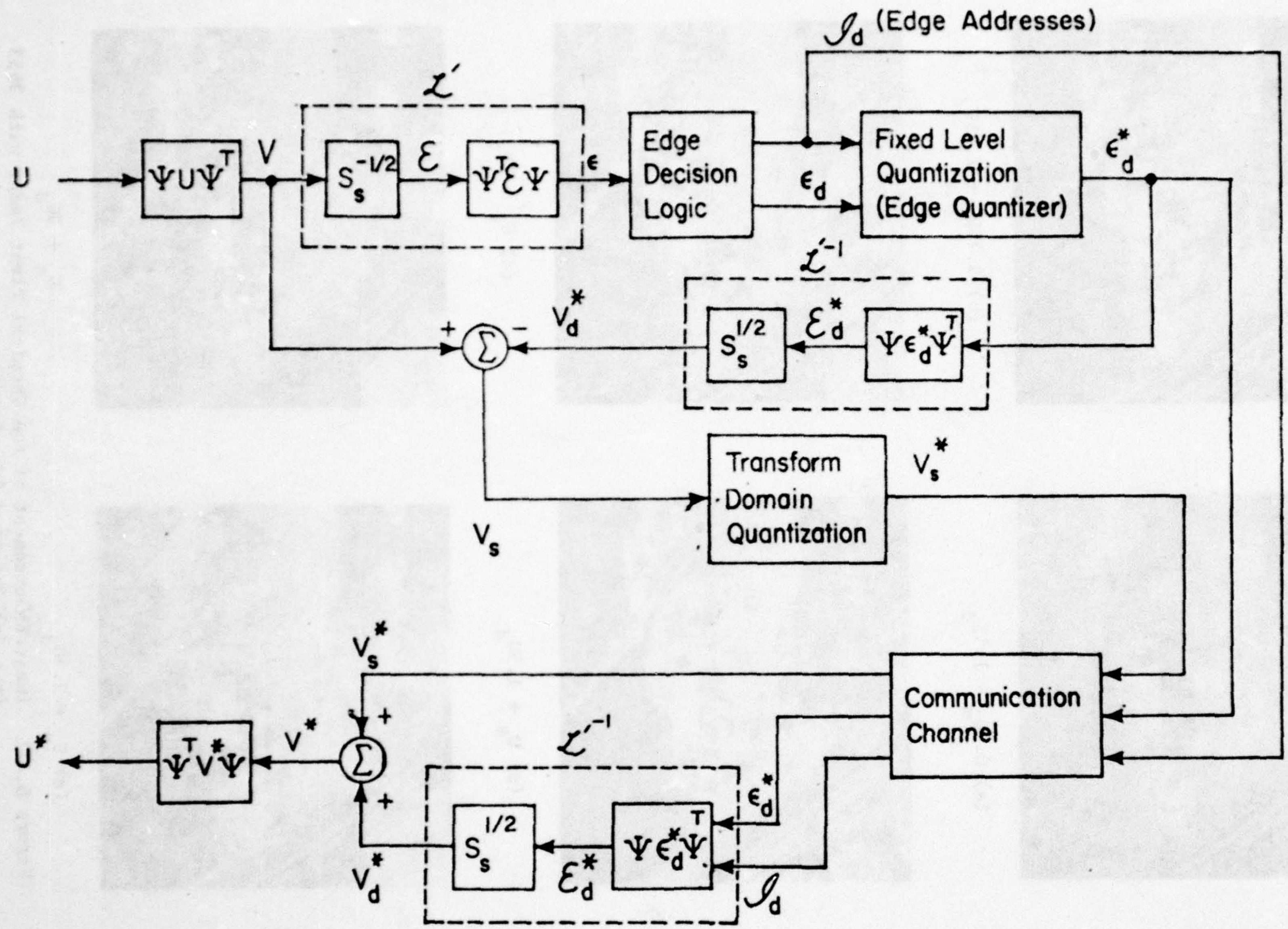


Figure 5.5 Transform Coding Based on Stochastic Segmentation

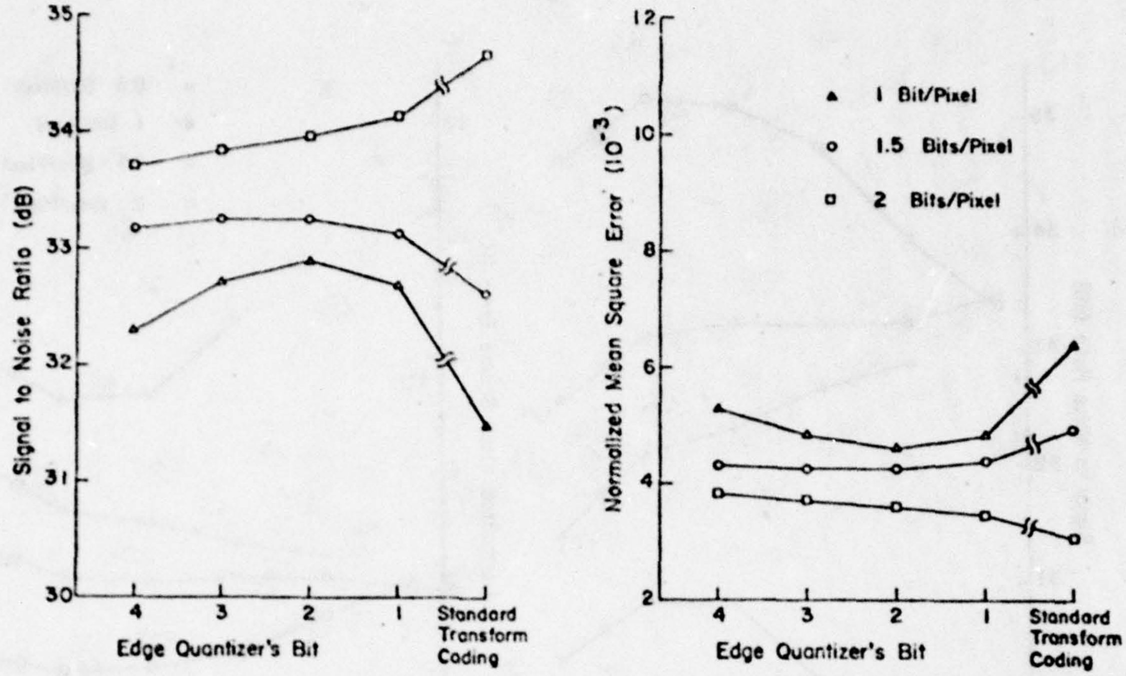


Figure 5.6 Feature Transform Coding Results for the Girl Image versus Various Edge Quantizer's Bits with 5737 ($t=1.5\sigma$) Detected Edge Points

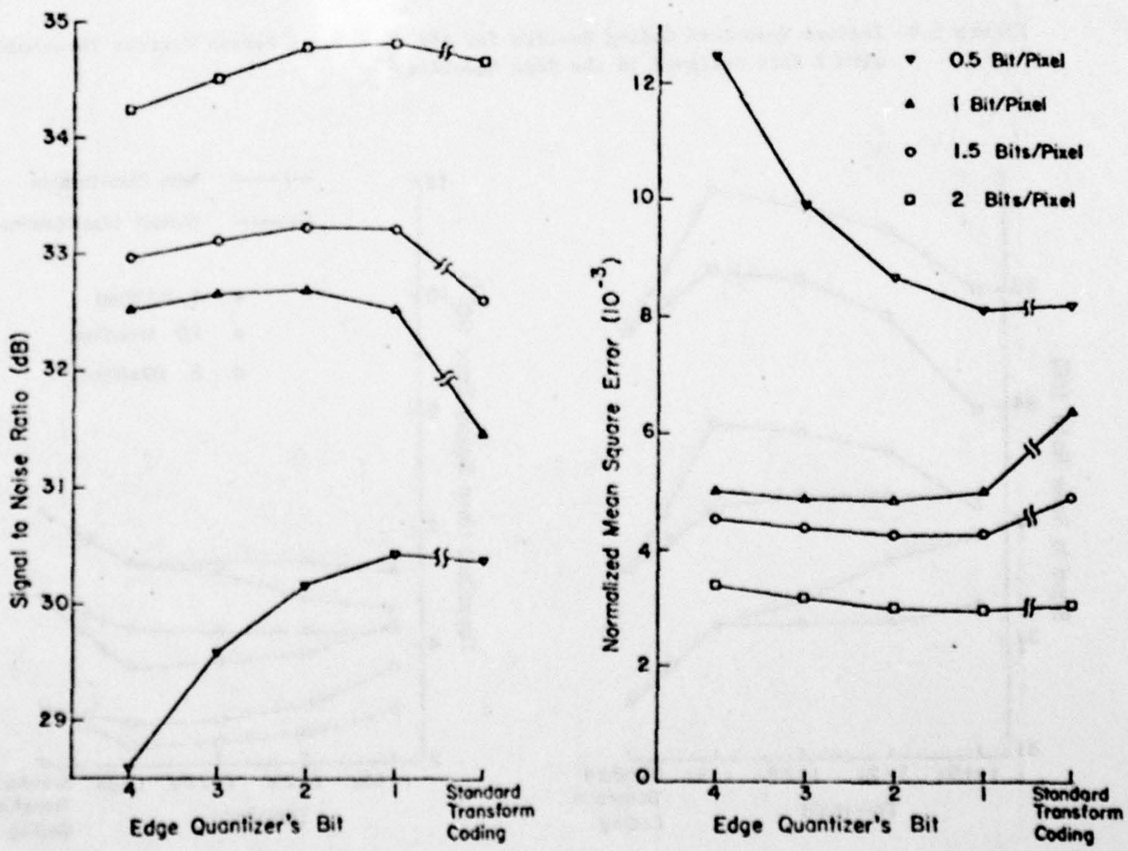


Figure 5.7 Feature Transform Coding Results for the Girl Image versus Various Edge Quantizer's Bits with 3111 ($t=2\sigma$) Detected Edge Points.

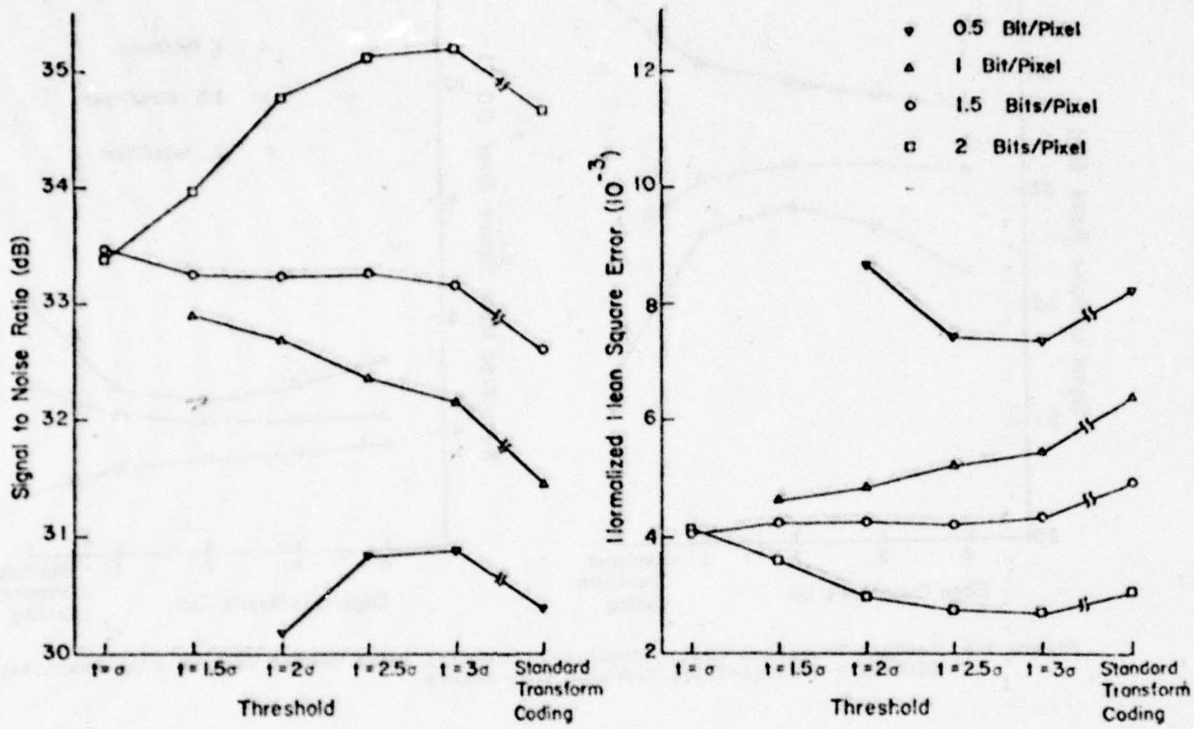


Figure 5.8 Feature Transform Coding Results for the Girl Image versus Various Thresholds with 2 Bits Assigned to the Edge Quantizer

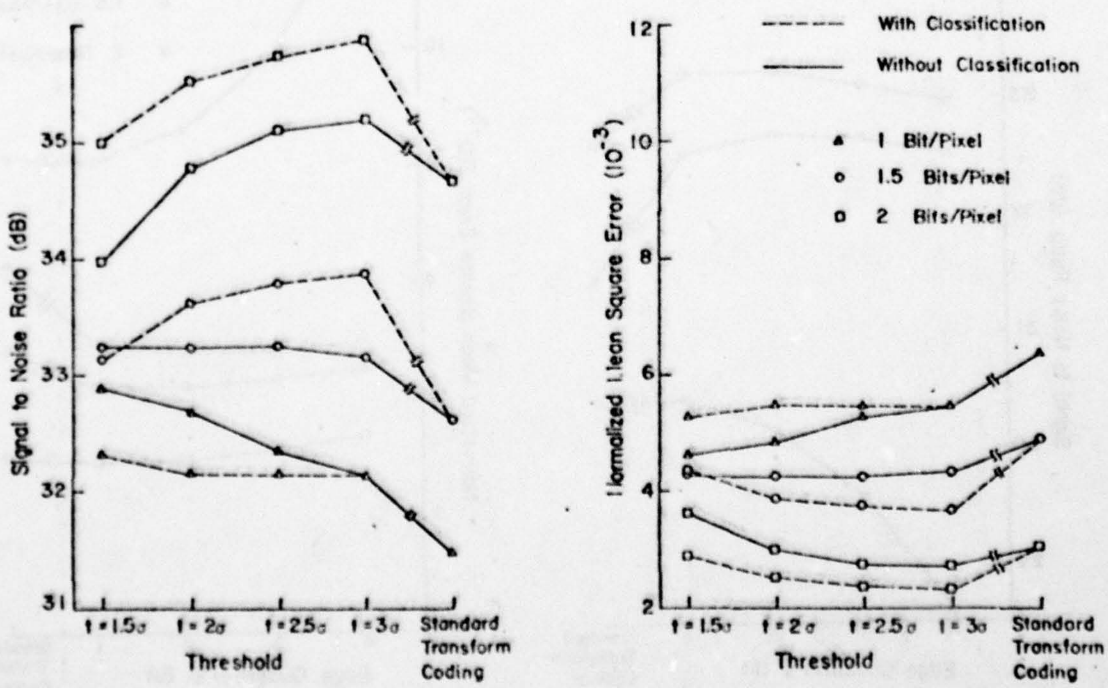


Figure 5.9 Classification Feature Transform Coding Results for the Girl Image versus Various Thresholds with 2 Bits Assigned to the Edge Quantizer

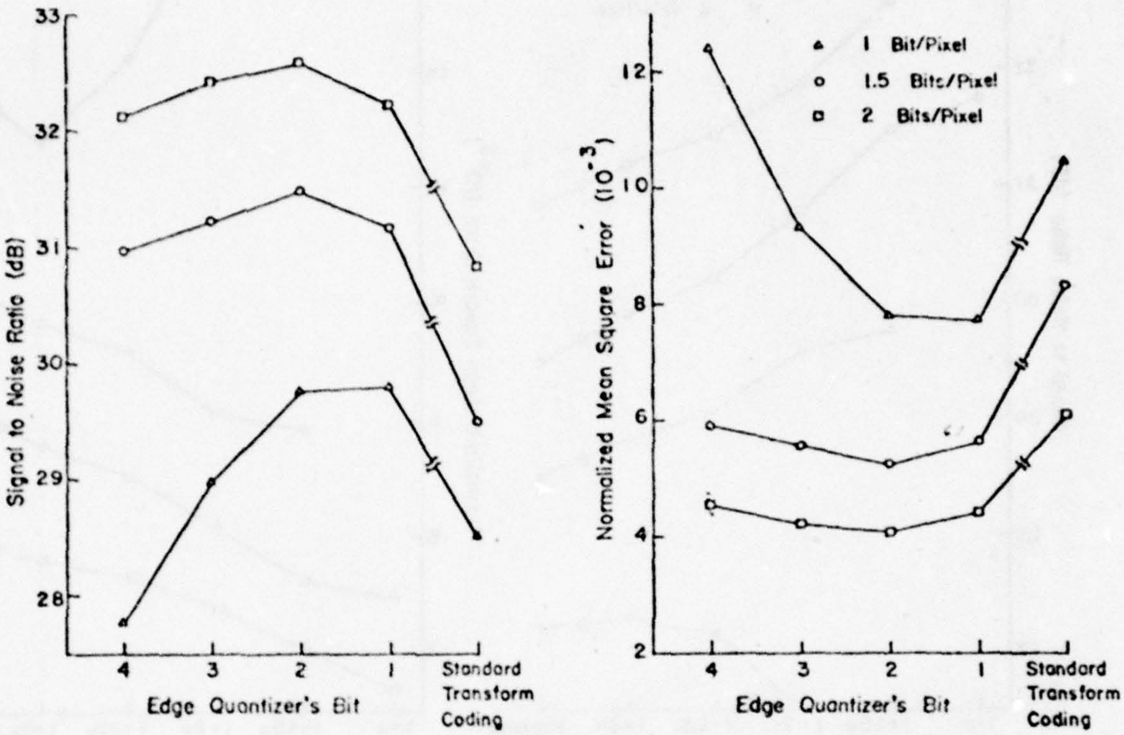


Figure 5.10 Feature Transform Coding Results for the Chemical Plant Image versus Various Edge Quantizer's Bits with 7079 (t = 1.5σ) Detected Edge Points

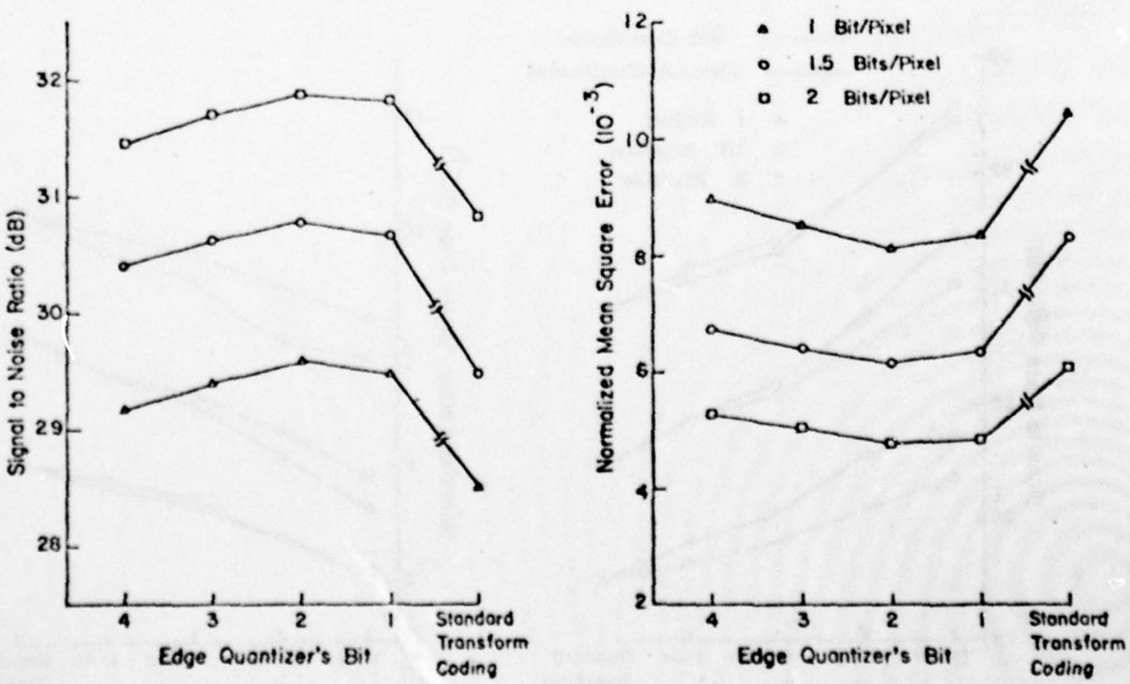


Figure 5.11 Feature Transform Coding Results for the Chemical Plant Image versus Various Edge Quantizer's Bits with 3653 (t = 2σ) Detected Edge Points

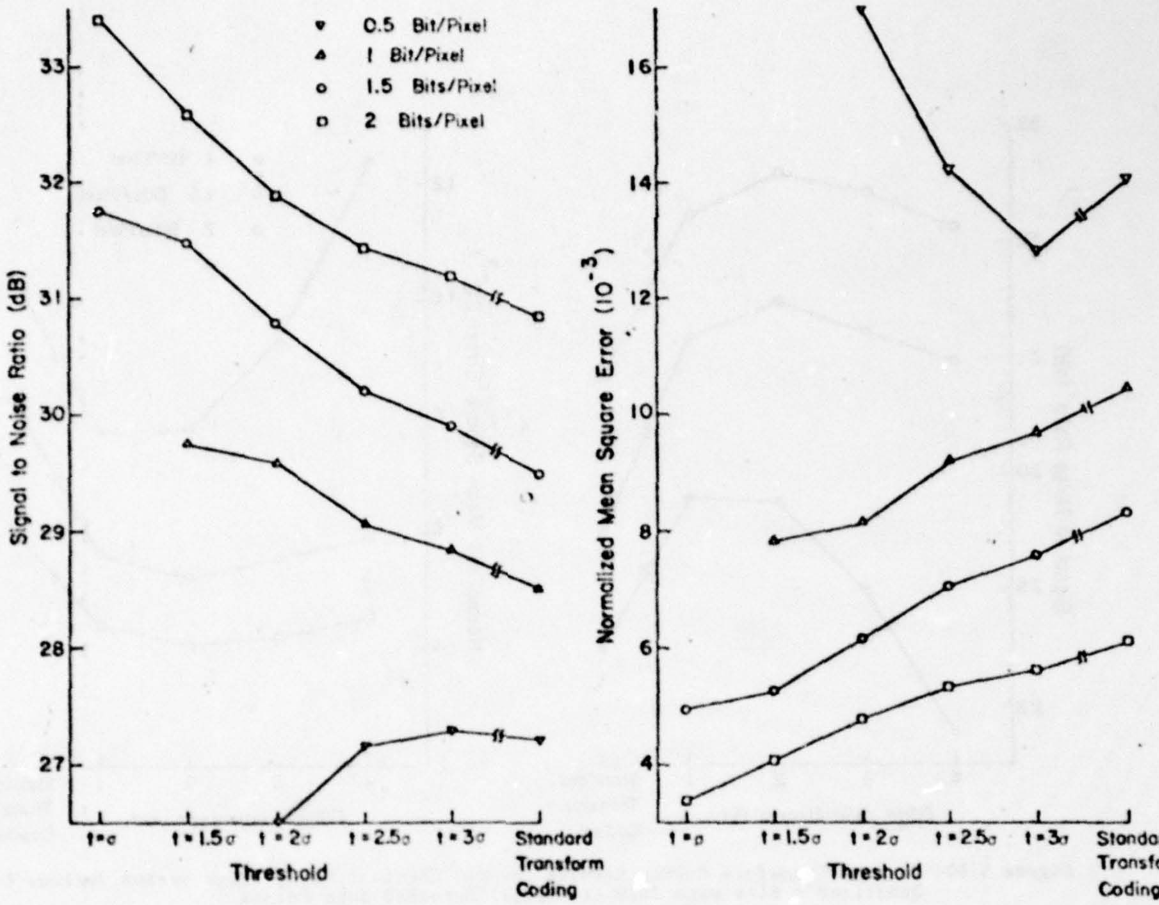


Figure 5.12 Feature Transform Coding Results for the Chemical Plant Image vs. Various Thresholds with 2 Bits Assigned to the Edge Quantizer

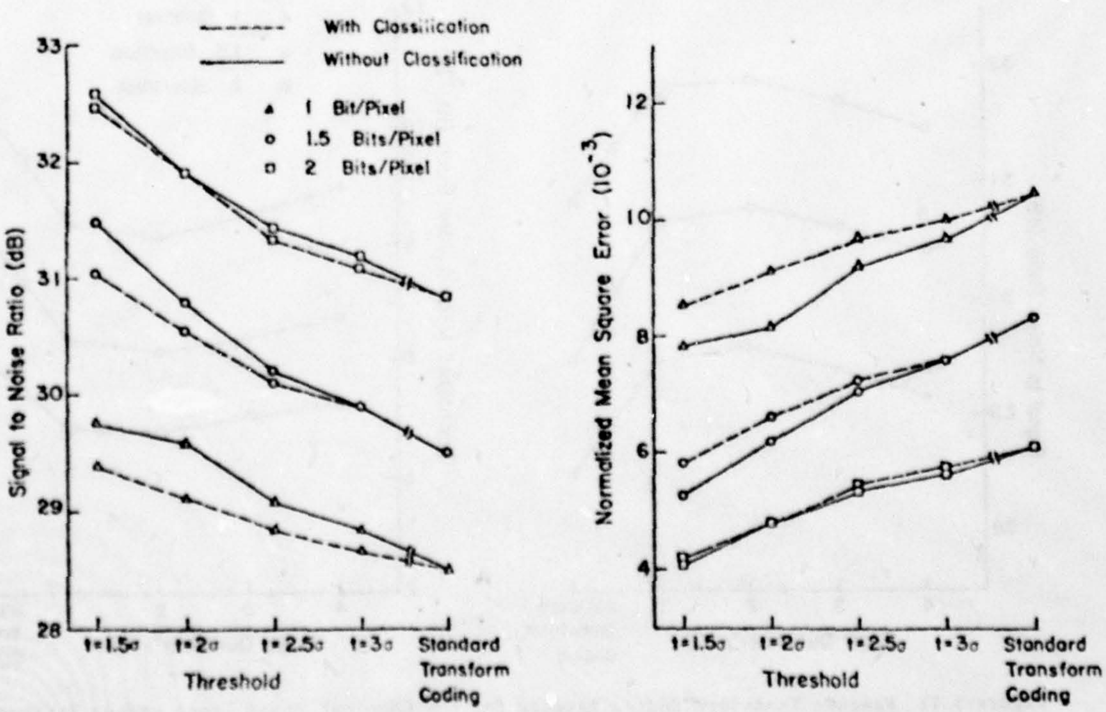


Figure 5.13 Classification Feature Transform Coding Results for the Chemical Plant Image vs. Various Thresholds with 2 Bits Assigned to the Edge Quantizer

AD-A078 781 CALIFORNIA UNIV DAVIS SIGNAL AND IMAGE PROCESSING LAB F/G 9/4
APPLICATIONS OF STOCHASTIC MODELS FOR IMAGE DATA COMPRESSION.(U)
SEP 79 S WANG , A K JAIN DAAG29-78-G-0206

UNCLASSIFIED ARO-16222.3-EL NL

3 OF 3
AD-
A078781

END
DATE
FILMED
1 - 80
DDC

(a) 1 Bit/Pixel,
Standard Transform

(d) 1 Bit/Pixel,
Feature Transform

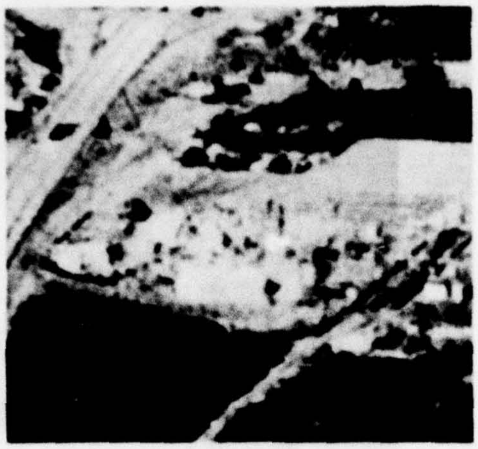
(b) 1.5 Bits/Pixel,
Standard Transform

(e) 1.5 Bits/Pixel,
Feature Transform

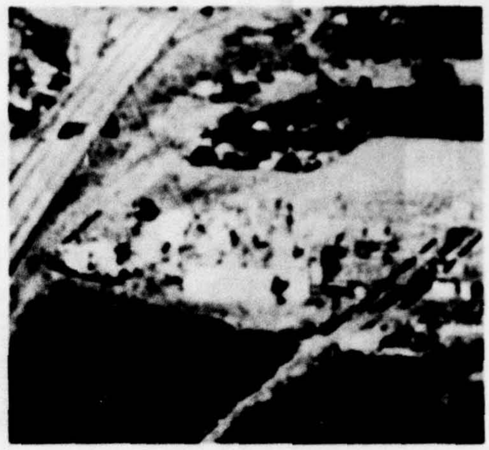
(c) 2 Bits/Pixel,
Standard Transform

(f) 2 Bits/Pixel,
Feature Transform

Figure 5.14 Comparison of Standard Transform Encoded and Feature Transform Encoded ($n_e = 2$ Bits, $t = 1.5\sigma$, 5737 Detected Edge Points) Girl Images



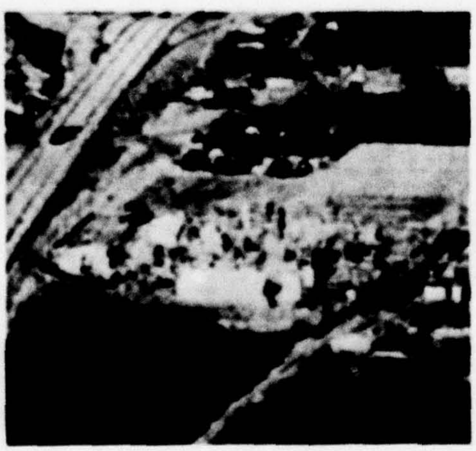
(a) 1 Bit/Pixel
Standard Transform



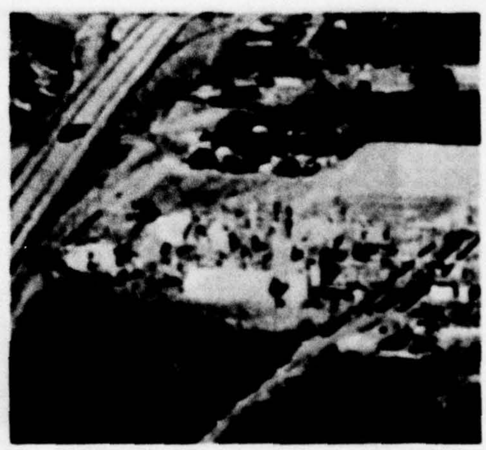
(d) 1 Bit/Pixel,
Feature Transform

(b) 1.5 Bits/Pixel,
Standard Transform

(e) 1.5 Bits/Pixel
Feature Transform



(c) 2 Bits/Pixel,
Standard Transform



(f) 2 Bits/Pixel,
Feature Transform

Figure 5.15 Comparison of Standard Transform Encoded and Feature Transform Encoded ($n_e = 2$ Bits, $t = 1.5\sigma$, 7079 Detected Edge Points) Chemical Plant Images

(a) 1 Bit/Pixel

(b) 1.5 Bits/Pixel

(c) 2 Bits/Pixel

(d) 1 Bit/Pixel, Classification

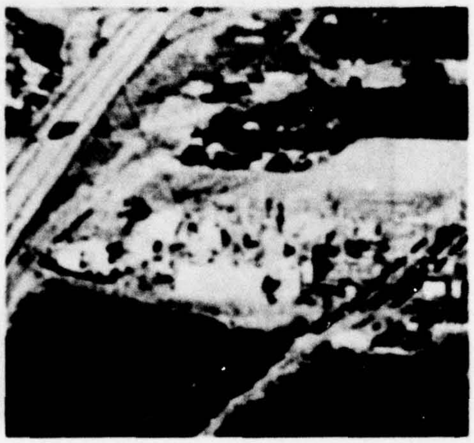
(e) 1.5 Bits/Pixel, Classification

(f) 2 Bits/Pixel, Classification

Figure 5.16 Feature Transform Encoded Girl Image with n_e = 2 Bits and $t = 1.5\sigma$ (5737 Detected Edge Points)

(a) 1 Bit/Pixel

(d) 1 Bit/Pixel, Classification



(b) 1.5 Bits/Pixel

(e) 1.5 Bits/Pixel, Classification

(c) 2 Bits/Pixel

(f) 2 Bits/Pixel, Classification

Figure 5.17 Feature Transform Encoded Chemical Plant Image with $n_e = 2$ Bits and $t = 1.5\sigma$ (7079 Detected Edge Points)

Table 5.1(a) Girl Image Thresholds and No. of Detected Edge Points

Threshold	No. of Edge Points	Entropy
$t = \sigma$	11624	3.190
$t = 1.5\sigma$	5737	3.626
$t = 2\sigma$	3111	3.949
$t = 2.5\sigma$	1768	3.957
$t = 3\sigma$	1063	3.698

Table 5.1(b) Chemical Plant Image Thresholds and No. of Detected Edge Points

Threshold	No. of Edge Points	Entropy
$t = \sigma$	14138	3.102
$t = 1.5\sigma$	7079	3.446
$t = 2\sigma$	3653	3.778
$t = 2.5\sigma$	1843	3.905
$t = 3\sigma$	923	3.585

Table 5.2 Feature Transform Coding Results for the Girl Image with 5737 ($t = 1.5\sigma$) Detected Edge Points

Average Rate	2		1.5		1	
Edge Quantizer's Bit	SNR(db)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)
$n_e = 4$	33.714	0.382	33.187	0.432	32.300	0.530
$n_e = 3$	33.844	0.371	33.252	0.426	32.710	0.482
$n_e = 2$	33.966	0.361	33.252	0.426	32.899	0.462
$n_e = 1$	34.133	0.347	33.126	0.438	32.687	0.485
Standard Xform Coding	34.662	0.308	2.614	0.493	31.480	0.640

Table 5.3 Feature Transform Coding Results for the Girl Image with 3111 ($t = 2\sigma$) Detected Edge Points

Average Rate	2		1.5		1		0.5	
Edge Quantizer's Bit	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)
$n_e = 4$	34.245	0.339	32.964	0.455	32.509	0.505	28.584	1.247
$n_e = 3$	34.511	0.318	33.121	0.438	32.659	0.488	29.583	0.990
$n_e = 2$	34.784	0.299	33.238	0.427	32.699	0.483	30.162	0.867
$n_e = 1$	34.814	0.297	33.213	0.429	32.543	0.501	30.441	0.813
Standard Xform Coding	34.662	0.308	32.614	0.493	31.480	0.640	30.396	0.821

Table 5.4 Feature Transform Coding Results for the Girl Image with 2 Bits Assigned to the Edge Quantizer

Average Rate	2		1.5		1		0.5	
Threshold	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)
$t = \sigma$	33.378	0.413	33.470	0.405	-	-	-	-
$t = 1.5\sigma$	33.966	0.361	33.252	0.426	32.899	0.462	-	-
$t = 2\sigma$	34.784	0.299	33.238	0.427	32.699	0.483	30.162	0.867
$t = 2.5\sigma$	35.127	0.276	33.261	0.425	32.358	0.523	30.839	0.742
$t = 3\sigma$	35.209	0.271	33.162	0.434	32.159	0.547	30.870	0.736
Standard Xform Coding	34.662	0.308	32.614	0.493	31.480	0.640	30.396	0.821

Table 5.5 Classification Feature Transform Coding Results for the Girl Image with 2 Bits Assigned to the Edge Quantizer

Average Rate	2		1.5		1	
Threshold	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)
$t = 1.5\sigma$	35.019	0.283	33.142	0.436	32.323	0.527
$t = 2\sigma$	35.537	0.251	33.632	0.390	32.163	0.547
$t = 2.5\sigma$	35.750	0.239	33.802	0.375	32.149	0.549
$t = 3\sigma$	35.873	0.233	33.875	0.369	32.148	0.549
Standard Xform Coding	34.662	0.308	32.614	0.493	31.480	0.640

Table 5.6 Feature Tansform Coding Results for the Chemical Plant Image with 7079 ($t = 1.5\sigma$) Detected Edge Points

Average Rate	2		1.5		1	
Edge Quantizer's Bit	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)
$n_e = 4$	32.123	0.454	30.971	0.592	27.761	1.240
$n_e = 3$	32.474	0.424	31.236	0.557	28.986	0.936
$n_e = 2$	32.586	0.408	31.482	0.527	29.758	0.783
$n_e = 1$	32.228	0.443	31.170	0.566	29.800	0.776
Standard Xform Coding	30.830	0.611	29.495	0.832	28.508	1.044

Table 5.7 Feature Transform Coding Results for the Chemical Plant Image with 3653 ($t = 2\sigma$) Detected Edge Points

Average Rate	2		1.5		1		0.5	
Edge Quantizer's Bit	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)
$n_e = 4$	31.457	0.530	30.405	0.675	29.169	0.897	24.372	2.706
$n_e = 3$	31.697	0.504	30.624	0.642	29.395	0.851	25.562	2.058
$n_e = 2$	31.885	0.480	30.791	0.617	29.592	0.814	26.394	1.699
$n_e = 1$	31.836	0.485	30.666	0.635	29.479	0.835	26.795	1.549
Standard Xform Coding	30.830	0.611	29.495	0.832	28.508	1.044	27.214	1.407

Table 5.8 Feature Transform Coding Results for the Chemical Image with 2 Bits Assigned to the Edge Quantizer

Average Rate	2		1.5		1		0.5	
Threshold	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)
$t = \sigma$	33.402	0.338	31.751	0.495	-	-	-	-
$t = 1.5\sigma$	32.586	0.408	31.482	0.527	29.758	0.783	-	-
$t = 2\sigma$	31.885	0.480	30.791	0.617	29.592	0.814	26.394	1.699
$t = 2.5\sigma$	31.428	0.533	30.208	0.706	29.062	0.919	27.169	1.421
$t = 3\sigma$	31.189	0.563	29.894	0.759	28.837	0.968	27.291	1.382
Standard Xform Coding	30.830	0.611	29.495	0.832	28.508	1.044	27.214	1.407

Table 5.9 Classification Feature Transform Coding Results for the Chemical Plant Image with 2 Bits Assigned to the Edge Quantizer

Average Rate	2		1.5		1	
Threshold	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)	SNR(dB)	NMSE(%)
$t = 1.5\sigma$	32.461	0.420	31.044	0.582	29.392	0.852
$t = 2\sigma$	31.902	0.478	30.546	0.661	29.106	0.910
$t = 2.5\sigma$	31.334	0.545	30.095	0.725	28.832	0.969
$t = 3\sigma$	31.083	0.577	29.893	0.759	28.669	1.066
Standard Xform Coding	30.830	0.611	29.495	0.832	28.508	1.044

CHAPTER SIX

SUMMARY AND RECOMMENDATIONS FOR FUTURE RESEARCH

This thesis has shown that the application of the stochastic image models based on PDEs to the problem of image data compression has achieved considerable success. The use of image models has shown the connection between predictive, hybrid and transform coding schemes. From the choice of an approximate image model which gives the best approximation to the actual statistics of a class of images being processed, on would be able to determine the most efficient data compression scheme.

Simple basis restriction error experiments were performed to compare the energy compaction property of various image models. The results show that both the noncausal and semicausal models give superior performance over the causal models. Evidently, different spectral density function shapes of image models were the dominating factor in this aspect of study.

The performance of the hybrid coding scheme was significantly improved by two very simple methods, i.e., (1) Adaptive Variance Estimation: Adapting the variance of the prediction error, in each DPCM channel, the updated variance was used to adjust the spacing of the quantizer levels. (2) Adaptive Classification: Adapting the bit rate to the local variance of each image column.

The adaptive variance estimation scheme was shown having the potential of maintaining the performance of a hybrid coder, designed for a nominal statistics, in the face of changing statistics. The adaptive classification scheme performed more effectively at low bit

rate and was robust with respect to channel errors. Hybrid coding of noisy images was achieved by replacing the predictor in the DPCM loop by a Kalman filter. Experimental results showed that the two bit rate quantizer gave a restored image with SNR slightly less than the infinite rate (without quantizer in the DPCM loop).

A new technique called feature transform coding for efficient transform coding of images was presented. The experimental results of applying this technique to actual image data (Girl and Chemical Plant images) show that it provides us a flexible way of getting a good visual image (subject to a fixed bit rate) while maintaining the mean square error performance. It is evidenced by referring to Figures 5.6 - 5.13. The incorporation of classification, a scheme analogous to the adaptive classification hybrid coding, results in further improvement in the performance for simple structured images at bit rate $\geq$ 1 bit/pixel. Moreover, its complexity over the feature transform coder is marginal.

From a practical point of view, the most important results of this research is that it provides a theoretical framework for optimizing the design of an intraframe coding system for two dimensional data modeled by random fields. The various coding algorithms developed here can be easily implemented for designing practical coding systems for images.

Areas in need of further investigation are:

1) Inclusion of NC2 and NC3 models to the design of feature transform coder. Since only NC1 model based feature transform coder has been fully explored.

2) Extension to the higher order stochastic image models which might be needed to represent image spectral density functions (SDFs) more accurately.

3) Application of data compression schemes to biomedical images.

4) Study the effects of incorporating variance estimation scheme in the feature transform coder.

5) Use of stochastic models in the design of coders in the presence of channel errors.

APPENDIX A

QUANTIZER CONSIDERATIONS

A.1 Introduction

The basic principle in designing a quantizer is to determine the optimum quantization levels so as to minimize the overall mean square error between the original signal and quantized signal. The exact approach to quantization problem can only be solved by wasting a considerable amount of computing time and gives little insight for someone who has a new quantization problem and needs it to be solved in a small period of time. The useful approximation to the optimum quantizer was first proposed by Algazi [2]. In this Appendix, we have modified his formulas slightly and tabulated the characteristics of the resulting quantizer (so called compandor) for Gaussian and Laplacian distributed signals together with the optimum nonuniform (in the minimum mean square error sense) and optimum uniform quantizers which have been considered by Max [44] and others. Also rate-distortion functions f(n) are listed which were obtained by a piecewise exponential curve fitting technique to the tabulated quantizer numerical results.

A.2 Quantizer Structures

A typical quantizer characteristic is shown in Fig. A.1. The quantizer output y is assigned to one of N discrete values (Fig. A.1(a) and Fig. A.1(b) are for N even and odd respectively) depending on the value of input x. The x_i and y_i are called the "decision levels" and "quantization levels" respectively. If the probability density function, p(x), of the quantizer input is given and is assumed to be symmetric, one would

have different quantizers according to the diverse assumptions of design constraints. In this section, we briefly review some of the useful quantizer reults. The interested reader can find detailed information in references [2,40,44,51,72]

A.2.1 Optimum Nonuniform Quantizer

In this quantizer, the design procedure is to find the N y_i's and associated x_i's in Fig. A.1 so as to minimize the distortion, D, which is defined as the expected value of $W(\varepsilon)$, where W is an error weighting function and ε is the quantization error. For simplicity, $W(\varepsilon)$ is assumed to be ε^2, the square value of the quantization error. The distortion is obtained as

Case 1: N = even

$$D = 2 \sum_{i=1}^{N/2} \int_{x_{i-1}}^{x_i} (y_i - x)^2 p(x)\,dx \qquad \text{(A.2.1-1)}$$

where $x_0 = 0$, $x_{\frac{N}{2}} = \infty$, and y_i lies between x_{i-1} and x_i.

Case 2: N = odd

$$D = \int_{-x_1}^{x_1} (y_1 - x)^2 P(x)\,dx + 2 \sum_{i=1}^{\frac{N-1}{2}} \int_{x_i}^{x_{i+1}} (y_{i+1} - x) p(x)\,dx \qquad \text{(A.2.1-2)}$$

where $y_1 = 0$, $x_1 > 0$, $x_{\frac{N+1}{2}} = \infty$, and y_i lies between x_{i-1} and x_i.

In order to minimize D for a fixed N, the necessary conditions are obtained by differentiating D with respect to the x_i's and y_i's and setting

the results equal to zero. After simple numerical manipulation, we have

Case 1: N = even

$$x_i = \tfrac{1}{2}(y_i + y_{i+1}) \qquad i = 1, \ldots, \frac{N}{2} - 1 \qquad \text{(A.2.1-3a)}$$

$$\int_{x_{i-1}}^{x_i} (y_i - x) p(x) dx = 0 \qquad i = 1, \ldots, \frac{N}{2} \qquad \text{(A.2.1-3b)}$$

Case 2: N = odd

$$x_i = \tfrac{1}{2}(y_i + y_{i+1}) \qquad i = 1, \ldots, \frac{N-1}{2} \qquad \text{(A.2.1-4a)}$$

$$\int_{x_{i-1}}^{x_i} (y_i - x) p(x) dx = 0 \qquad i = 2, \ldots, \frac{N+1}{2} \qquad \text{(A.2.1-4b)}$$

The above equations mean that x_i lies midway between the adjacent quantization levels and y_i is the centroid of the area under $p(x)$ and lies between the adjacent decision levels.

A.2.2 Optimum Uniform Quantizer

This quantizer gives the minimum distortion under the constraint

$$x_i - x_{i-1} = y_i - y_{i-1} = q \quad \forall i \qquad \text{(A.2.2-1)}$$

where q is the equal length spacing between adjacent decision levels or quantization levels.

This is an easier problem to be solved than the previous optimum non-uniform quantizer. For this case, the distortion, D, becomes a function of q and of any particular value of N. If the probability density function

of the quantizer input is assumed to be known and symmetric, then a symmetric answer for the decision levels and quantization levels will be obtained. The distortion, D, can be obtained as

Case 1: N = even

$$D = 2\sum_{i=1}^{\frac{N}{2}-1}\int_{(i-1)q}^{iq}\left[\left(\frac{2i-1}{2}\right)q-x\right]^2 p(x)\,dx + 2\int_{\left(\frac{N}{2}-1\right)q}^{\infty}\left[\left(\frac{N-1}{2}\right)q-x\right]^2 p(x)\,dx \qquad \text{(A.2.2-2)}$$

Case 2: N = odd

$$D = \int_{-\frac{q}{2}}^{\frac{q}{2}} x^2 p(x)\,dx + 2\sum_{i=1}^{\frac{N-3}{2}}\int_{(i-\frac{1}{2})q}^{(i+\frac{1}{2})q}(iq-x)^2 p(x)\,dx + 2\int_{\left(\frac{N-2}{2}\right)q}^{\infty}\left[\left(\frac{N-1}{2}\right)q-x\right]^2 p(x)\,dx \qquad \text{(A.2.2-3)}$$

The optimum uniform quantizer step size q can be obtained by differentiating D with respect to q and setting the results equal to zero. We have

Case 1: N = even

$$\frac{dD}{dq} = 2\sum_{i=1}^{\frac{N}{2}-1}(2i-1)\int_{(i-1)q}^{iq}\left[\left(\frac{2i-1}{2}\right)q-x\right]p(x)\,dx + 2(N-1)\int_{\left(\frac{N}{2}-1\right)q}^{\infty}\left[\left(\frac{N-1}{2}\right)q-x\right]p(x)\,dx = 0 \qquad \text{(A.2.2-4)}$$

Case 2: N = odd

$$\frac{dD}{dq} = 2\sum_{i=1}^{\frac{N-3}{2}} i\int_{(i-\frac{1}{2})q}^{(i+\frac{1}{2})q}(iq-x)p(x)\,dx + (N-1)\int_{\left(\frac{N-2}{2}\right)q}^{\infty}\left[\left(\frac{N-1}{2}\right)q-x\right]p(x)\,dx = 0 \qquad \text{(A.2.2-5)}$$

In either case, the problem is quite acceptable to machine computation when $p(x)$ and N are specified.

A.2.3 Useful Approximation to the Optimum Nonuniform Quantizer [2]

Since a quantizer is often easier to implement, we consider the nonuniform quantizer as the case cade of two nonlinear devices and of a uniform quantizer as shown in Fig. A.2. With a given uniform quantizer and a given probability density function, $p(x)$, the two nonlinear devices $f(\cdot)$ and $g(\cdot)$ in Fig. A.2 are chosen so as to minimize the distortion, $D = E[W\{g(y)-x\}]$, where W as defined in section A.2.1 is an error weighting function. The distortion is made up of two parts:

1. The distortion D_n obtained within the range of the uniform quantizer.
2. The distortion in the tails D_T for which the nonlinear devices $f(\cdot)$ and $g(\cdot)$ are completely ineffective.

The analog system is shown in Fig. A.3. The quantization noise due to uniform quantizer is assumed to be independent of the input signal for a large number of quantization levels, the effect of which has been discussed in detail by many researchers. Referring to Fig. A.3, the two extreme points of the output of the first nonlinear device is a dominating factor in the determination of the overall distortion, in other words, D is a function of $f(x_{max})$ and $f(x_{min})$, and $f(x_{max})-f(x_{min}) = Nq$, q being the step size of the uniform quantizer. The output of the second nonlinear device, $g(y)$, is expressed in terms of n and $f(x)$ as

$$g(y) = g[f(x) + n] \cong g[f(x)] + ng'[f(x)] \qquad \text{(A.2.3-1)}$$

If the quantization error is small, a Taylor series expression of g(y) gives the above equation directly. And the expression for D_n can be written as

$$D_n = \int_{-\frac{q}{2}}^{\frac{q}{2}} \int_{x_{min}}^{x_{max}} W\{q[f(\alpha)+\beta] - \alpha\} p_x(\alpha) p_n(\beta) d\alpha d\beta \qquad \text{(A.2.3-2)}$$

The limits on the integrals are $-\frac{q}{2}$ to $\frac{q}{2}$ for the integral in β and x_{min} to x_{max} $(f(x_{max}) - f(x_{min}) = Nq)$ for the integral in α.

If we assume $W(\varepsilon) = \varepsilon^2$ and g(y) is sufficiently smooth and the quantization error is small, then following the development in [2], one would be able to obtain

$$D_n = \frac{1}{12N^2} \left[\int_{x_{min}}^{x_{max}} [p_x(\alpha)]^{\frac{1}{3}} d\alpha \right]^3 \qquad \text{(A.2.3-3)}$$

If we take into account the probability of occurence of D_n and D_T, we have

$$D = D_n \left[1 - \int_{-\infty}^{x_{min}} p_x(\alpha) d\alpha - \int_{x_{max}}^{\infty} p_x(\alpha) d\alpha \right] + \int_{-\infty}^{x_{min}} [(x_{min} + \frac{q}{2}) - x]^2 p_x(\alpha) d\alpha$$

$$+ \int_{x_{max}}^{\infty} [(x_{max} - \frac{q}{2})]^2 p_x(\alpha) d\alpha \qquad \text{(A.2.3-4)}$$

For a symmetric probability density function, we set $x_{max} = -x_{min} = X$, and

$$D = \frac{2}{3N^2} \left[\int_0^X [p_x(\alpha)]^{\frac{1}{3}} d\alpha \right]^3 \left[1-2\int_X^{\infty} p_x(\alpha) d\alpha \right] + 2 \int_X^{\infty} [(X-\frac{q}{2})-x]^2 p_x(\alpha) d\alpha \qquad \text{(A.2.3-5)}$$

Equation (A.2.3-5) has to be minimized by proper choice of X. Once X is determined, then it can be shown that

$$f(x) = \frac{Nq\int_0^x [p_x(\alpha)]^{\frac{1}{3}}d\alpha}{2\int_0^X [p_x(\alpha)]^{\frac{1}{3}}d\alpha} \quad . \qquad (A.2.3\text{-}6)$$

It is noted from the above equation that the step size of the uniform quantizer will not affect the resulting nonuniform quantizer.

The characteristic of this quantizer for 2^m ($m = 1,\ldots,9$) quantization levels is given and discussed in the next section. We mention here that the derivation given in [2] erred in assuming the two extremes of the uniform quantizer shown in Fig. A.3 to be $\pm\left(\frac{N-1}{2}\right)q$ instead of $\pm\left(\frac{N}{2}\right)q$.

A.3 Results and Comparisons

The structures for the optimum uniform, optimum nonuniform and approximate nonuniform (compandor) quantizers are described in the previous sections. In this section, we present some nice characteristics of the approximate nonuniform quantizer along with the optimum uniform and optimum nonuniform quantizers. The equations defining the optimum nonuniform quantizer yield closed-form solution only in certain special cases, depending solely on the probability density function of the quantizer input signal. For example, even for the Gaussian distributed signal, for lack of the closed-form solution for the x_i (decision levels) and y_i (reconstruction levels), some sort of numerical solution must be sought. Max [44] suggested an iterative procedure to calculate x_i and y_i and

tabulated the values of x_i and y_i for N (number of levels) up to 36, since the time required to determine the quantizer structure increases exponentially with the total number of levels of the quantizer, Max's [44] work cannot be extended beyond N = 36. Kurtenbach and Wintz [40] applied the same numerical approach and tabulated the value of x_i and y_i to N = 512 ($N = 2^n$, n = 9, n : number of quantizer bits) for the Gaussian distributed signal. They confessed also that the determination of the quantizer structure for m = 6, 7, 8, 9 is limited because of the computer time required for each case. Paez and Glisson [51] published the quantizer results for the Laplacian and Gamma distributed signals facing the same kind of difficulty, and tabulated the data only up to N = 32. So the exact approach to the optimum nonuniform quantizer gives little insight and sometimes prsents someone who has a new quantization problem with a considerable amount of computation time to obtain a significant result. The second advantage of designing the approximate nonuniform quantizer is that it is easily implementable with only a marginal increase in its complexity compared to the optimum uniform quantizer.

Tables A.1 and A.2 list the approximate average amount of CDC CYBER 173 computer time required to determine the quantizer structure for the optimum nonuniform, optimum uniform and approximate nonuniform quantizers. A comparison of these three quantizers can be easily made from the entries of Tables A.1 and A.2 to conclude that the approximate nonuniform quantizer is the best choice as far as computation time is concerned.

Note that we have used our own numerical approach to solve the values of x_i and y_i for a given quantizer. The procedure is described as follows:

If the probability density function is known, we make an initial guess for x_i $(i = 1,\ldots,\frac{N}{2} - 1)$, where the x_i are uniformly spaced. $X_{\frac{N}{2}}$ is assumed to be equal to five times the standard deviation of the input signal. Then y_i is calculated $(i = 1,\ldots,\frac{N}{2})$ by solving (A.2.1-3b). x_i $(i = 1,\ldots,\frac{N}{2} - 1)$ is updated, denoted as $\hat{x}_i$, by using Newton's numerical method on (A.2.1-3a) compute resulting error $e = x_i - \hat{x}_i$. If it satisfies the required error bound, then x_i was chosen correctly, otherwise the process is repeated. Fig. A.4 shows the flow chart of this numerical procedure.

The correctness of the computer program was checked by printing out the numerical value of each data entry to five digits after the decimal point. Comparing the results with Max's [44], we have precise value for each x_i and y_i, and furthermore our accuracy is extended to five digits instead of four digits after the decimal point.

As a by-product of determining the quantizer structures, most of the useful information quantities were obtained. There are (a) the mean square error (M.S.E.) introduced in the quantization process, (b) the entropy at the quantizer output, (c) the signal to noise ratio, (d) the rate distortion function or source information rate. In addition, all the data entries tabulated are constructed for quantizer input signal with unit standard deviation (r.m.s. value). To obtain the 'decision levels' x_i or "quantization levels' y_i for signals with standard deviation different from unity, we simply need multiply the given values of data entries by the actual standard deviation. The M.S.E. is found by multiplying the given M.S.E. by the variance of the signal.

In all the measurements, after obtaining the values of x_i and y_i, the mean square errors (M.S.E.) are calculated by (A.2.1-1), (A.2.2-2) and (A.2.3-5) for optimum nonuniform, optimum uniform and approximate nonuniform quantizer respectively. Here we assume that the distribution of the quantizer input signal and quantizer structure is symmetric and that the total number of quantizer output levels is even. This constraint is not restrictive. For example, we have plotted Tables A.10-A.15 for $N = 2,\ldots,36$, which has both cases N = even and N = odd.

For entropy it is possible to treat the quantizer output as a discrete data source and can be calculated from the following formula.

$$\text{Entropy} = -2 \sum_{i=1}^{\frac{N}{2}} p_i \log_2 p_i \text{ bits per sample} \tag{A.3-1}$$

$$\text{where } p_i = p\,[y=y_i] = p\,[x_{i-1} \le x \le x_i] = \int_{x_{i-1}}^{x_i} p(x)dx \quad i = 1,\ldots,\frac{N}{2}.$$

Gaussian and Laplacian signal distributions were considered. These are the two commonly used assumptions for the signal distribution in the image processing area.

$$\text{Guassian density:} \quad p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{A.3-2}$$

$$\text{rms value} = \sigma$$

$$\text{Laplacian density:} \quad p(x) = \frac{\alpha}{2} e^{-\alpha|x|} \tag{A.3-3}$$

$$\text{rms value} = \sigma = \frac{\sqrt{2}}{\alpha}$$

The exact formula for p_i in either case can be expressed as follows:

Laplacian distributed:

$$P_i = \int_{x_{i-1}}^{x_i} p(x)dx = \int_{x_{i-1}}^{x_i} \frac{\alpha}{2} e^{-\alpha|x|} dx = -\tfrac{1}{2}\left[e^{-\alpha x_i} - e^{-\alpha x_{i-1}}\right] \quad i = 1, \ldots, \frac{N}{2}$$

(A.3-4)

Gaussian distributed:

$$P_i = \int_{x_{i-1}}^{x_i} p(x)dx = \int_{x_{i-1}}^{x_i} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx = \tfrac{1}{2}\left[\mathrm{erf}(x_i/\sqrt{2}\sigma) - \mathrm{erf}(x_{i-1}/\sqrt{2}\sigma)\right] \dagger$$

$$i = 1, \ldots, \frac{N}{2}$$

. (A.3-5)

The signal to noise ratio (S.N.R.) is defined as

$$\text{S.N.R.(dB)} = 10 \log_{10} \frac{1}{\text{mean square error (M.S.E.)}} \quad \text{(A.3-6)}$$

where we assume unity variance for the quantizer input signal.

Tables A.3 - A.8 tabulate the characteristics of three different quantizers with Gaussian and Laplacian distributed quantizer input signal.

In order to better compare the performances of these three quantizers, we have plotted the curves of all three quantizers into one single graph according to differene categories, i.e., mean square error, signal to noise ratio, and entropy. These are shown in Figs. A.5 - A.10. It is seen

† We use the National Bureau of Standards definition for the error function, i.e.,

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

from these figures that approximate nonuniform quantizers (compandor) perform virtually as good as optimum nonuniform quantizers, since the characteristic curves of these two quantizers are so close together. Fig. A.11 shows the unequal probabilities of quantizer output levels for the case of $N = 8$. Again, the similarity of the compandor and the nonuniform quantizer can be easily observed. Another significant result that can be observed from Figs. A.5 - A.10 is that the advantage of the optimum nonuniform quantizer and approximate nonuniform quantizer (compandor) over the optimum uniform quantizer starts with $n = 2$ and increases as n increases. This effect is due to the fact that both former quantizer structures have more quantization levels near the origin than the optimum uniform quantizer. Here we make the assumption that our quantizer structure is symmetric with respect to the origin. But the optimum uniform quantizer still has its own advantages. It is easily calculable for the x_i (decision levels) and y_i (quantization levels). Also Wood [72] showed that it gives the minimum distortion among the three quantizer structures if a fixed entropy of the quantizer output is required.

Since we have presented many tables for optimum nonuniform, optimum uniform and approximate nonuniform quantizers, all these tables are created under the assumption that we have a unity variance input signal. Using f(n) to denote M.S.E., there is no closed-form expression to relate f(n) and n (n:number of quantizer bits). However, one still can easily approximate the numerical results of f(n) by a piecewise exponential fitting numerical technique. Table A.9I shows the analytic models of f(n). If we denote $h(x) = f'^{-1}(x)$, the corresponding analytic models of h(x) are shown in Table A.9II. Tables A.9I and A.9II serve as a useful tool for solving the Rate-Distortion problem in information theory.

A.4 Conclusions

We have obtained the approximate nonuniform quantizer characteristics for Gaussian and Laplacian distributed signals and investigated the performance of this quantizer with optimum nonuniform quantizer as well as optimum uniform quantizer. We have also extended the performances of optimum nonuniform and optimum uniform quantizers to large quantization levels N = 512 ($N = 2^n : n = 9$) accompanied by useful information, viz., mean square error, signal to noise ratio and entropy, etc.

In practical implementation, approximate nonuniform quantizer seems as good as an optimum nonuniform quantizer, and it is easily calculable with only a marginal increase in its complexity compared to the optimum uniform quantizer. Also it is an acceptable and good quantizer structure if the computer system doesn't have enough buffer space to store the entire tabulated data for every large N or m.

The scheme described in this Appendix can be easily extended to other input signal processes and is not restricted to just Gaussian and Laplacian processes.

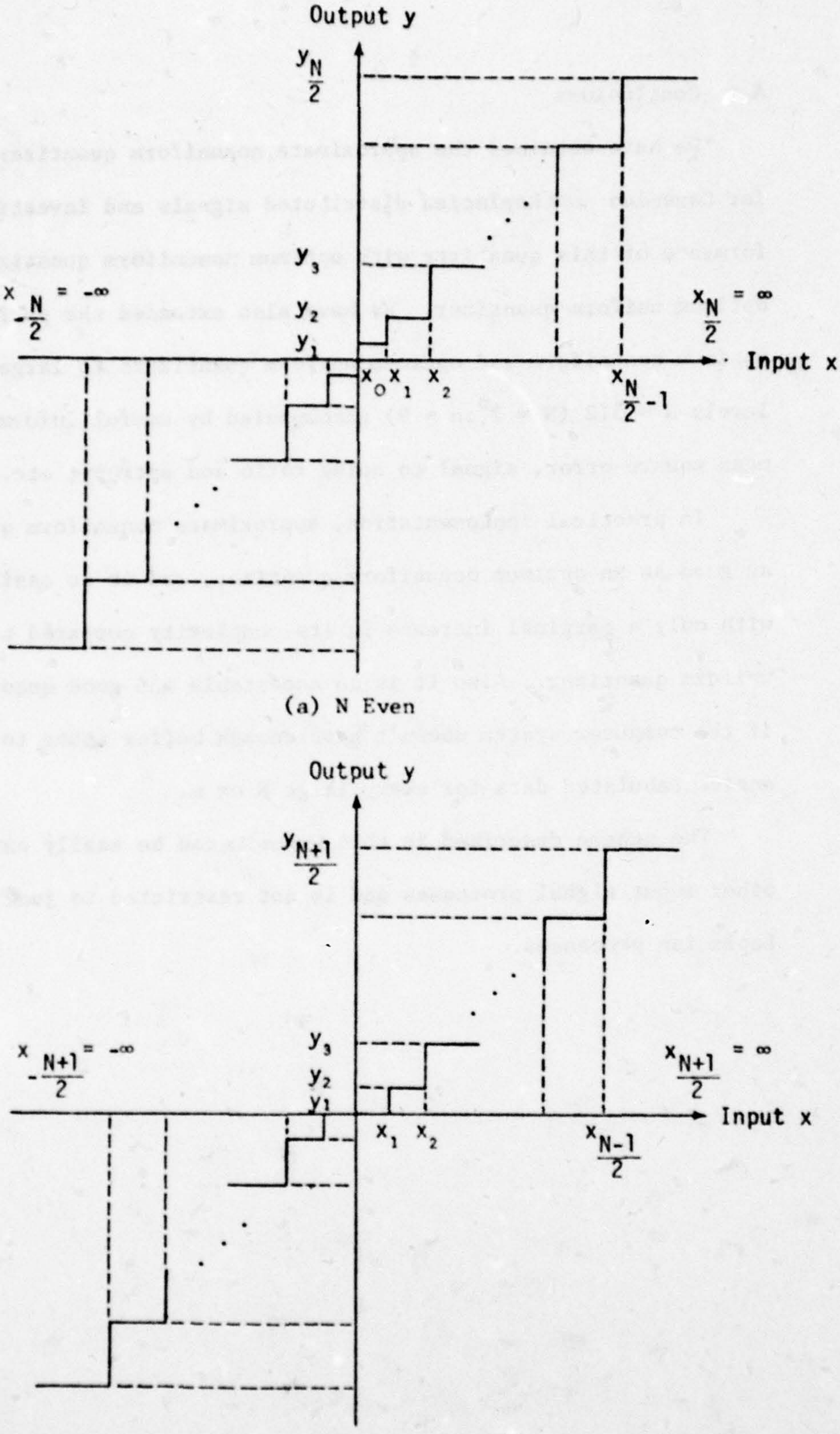


Figure A.1 Quantizer Characteristics

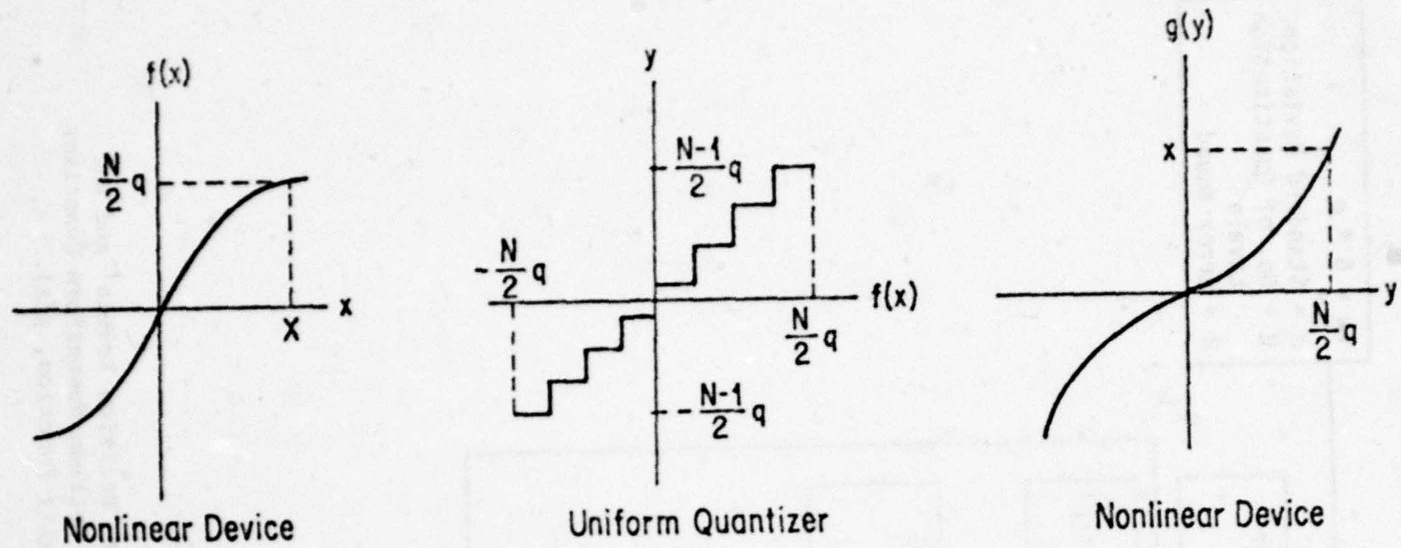


Figure A.2 Nonuniform Quantizer (Compandor)

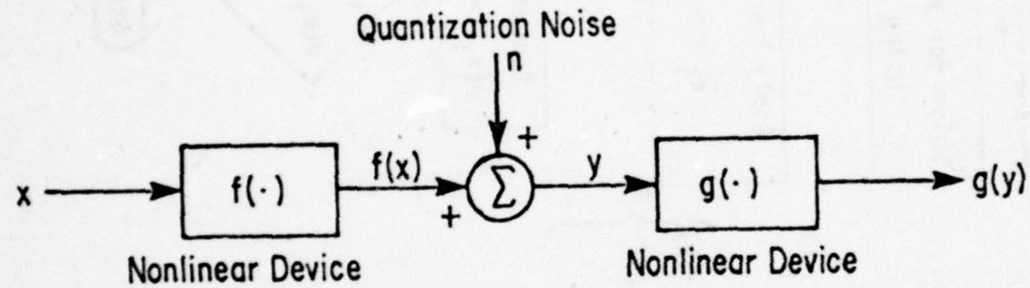


Figure A.3 Analog Model for Nonlinear Quantization

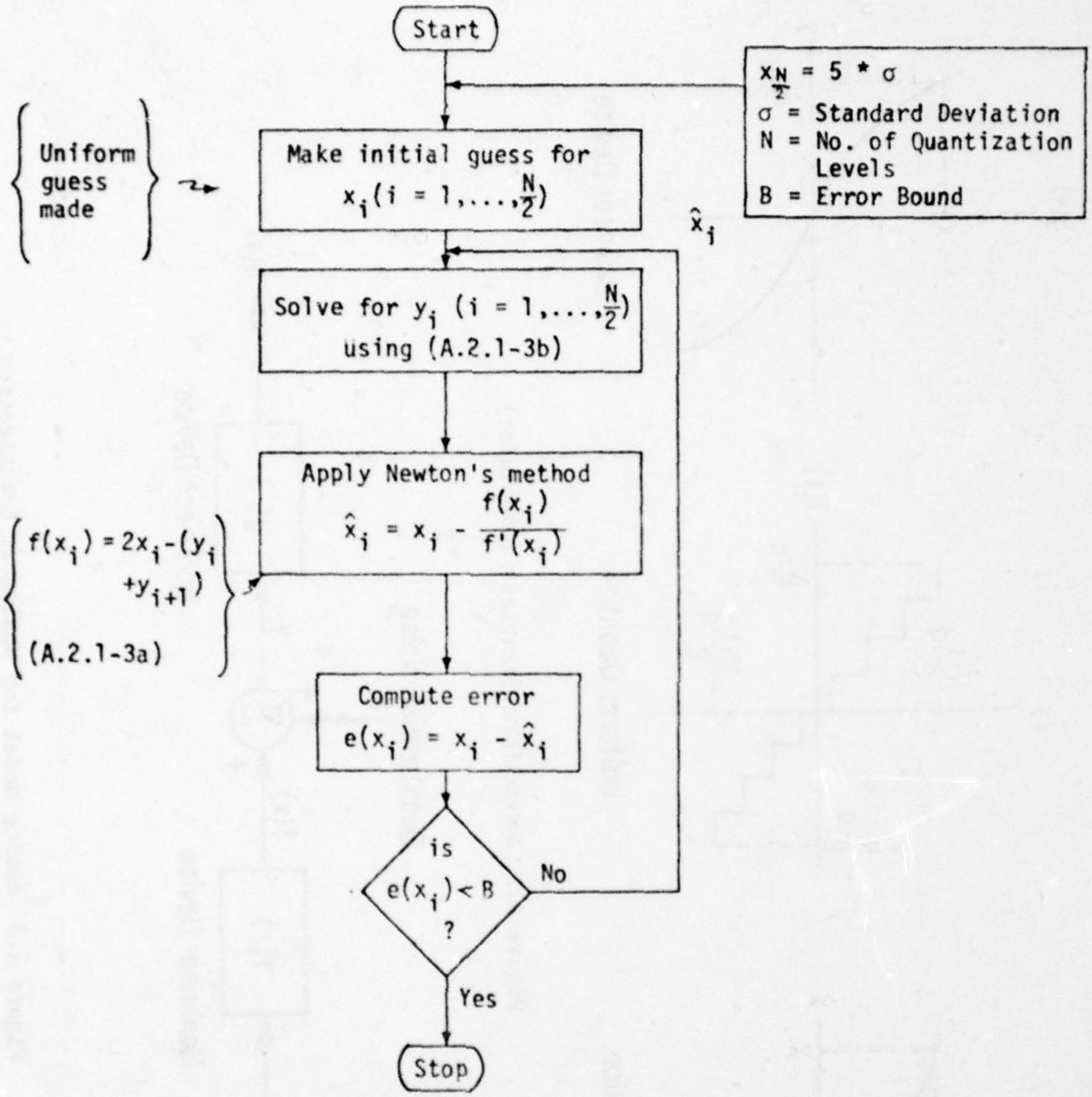


Figure A.4 Flow Chart of Determining the 'Decision Levels' and the 'Quantization Levels' for an Optimum Nonuniform Quantizer with a Given Probability Density Function, p(x).

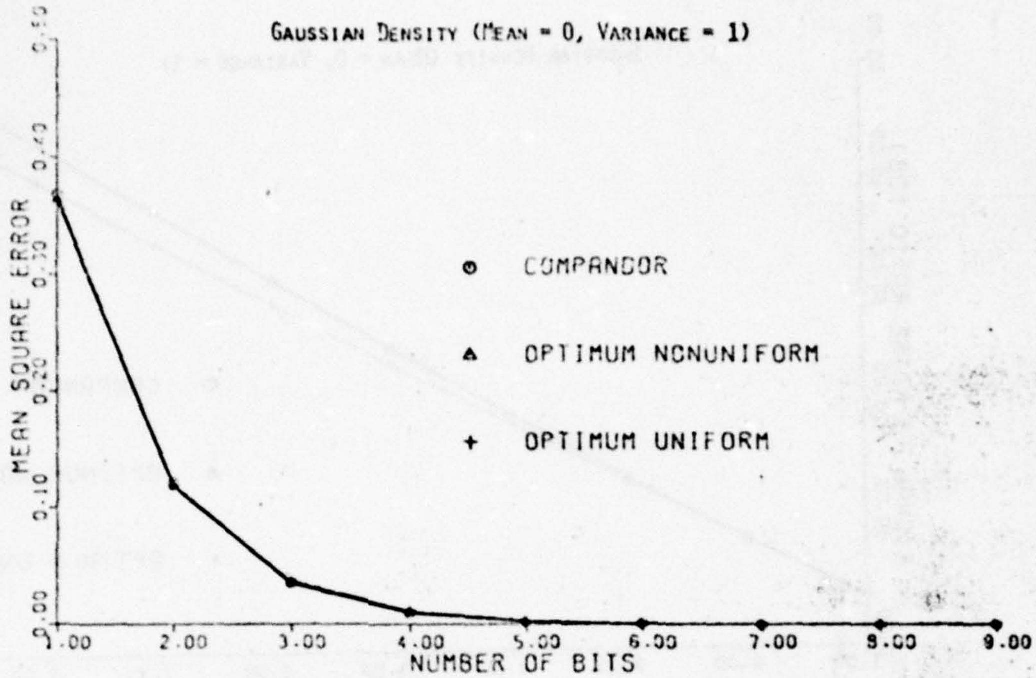


Figure A.5 Mean Square Error versus Number of Quantizer Bits for Signal with Gaussian Density (Mean = 0, Variance = 1)

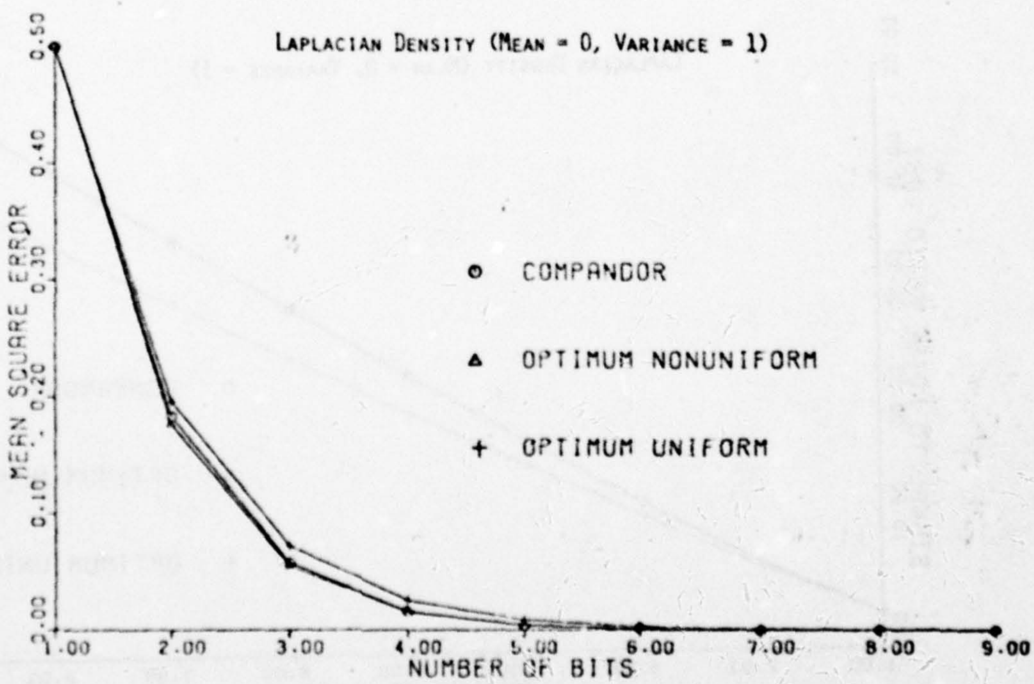


Figure A.6 Mean Square Error versus Number of Quantizer Bits for Signal with Laplacian Density (Mean = 0, Variance = 1)

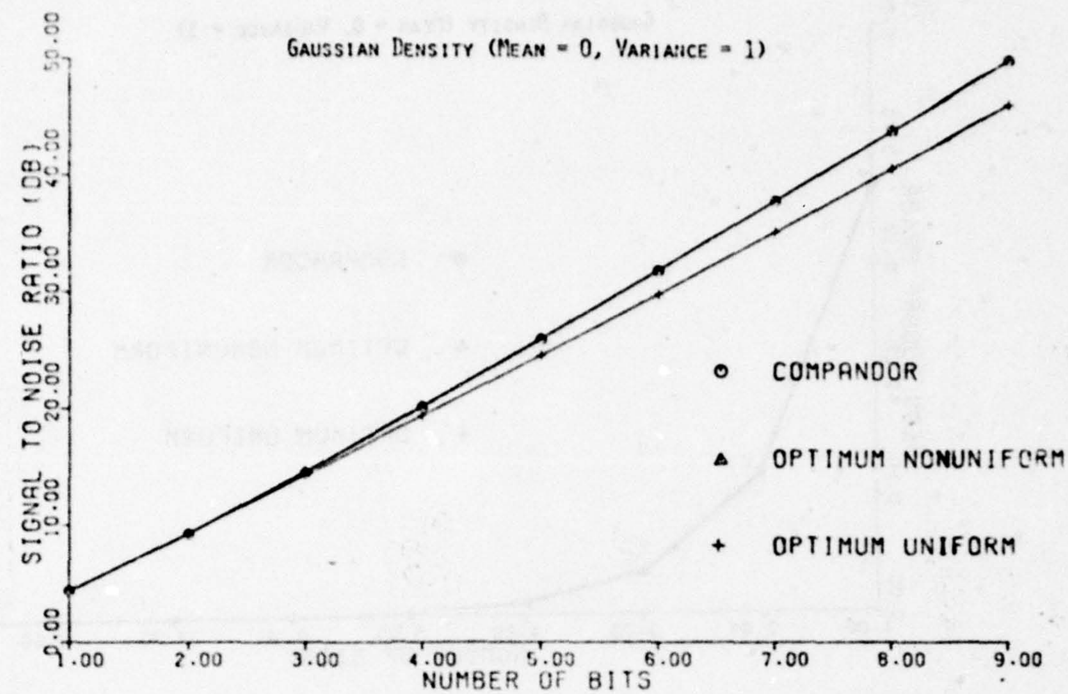


Figure A.7 Signal to Noise Ratio versus Number of Quantizer Bits for Signal with Gaussian Density (Mean = 0, Variance = 1)

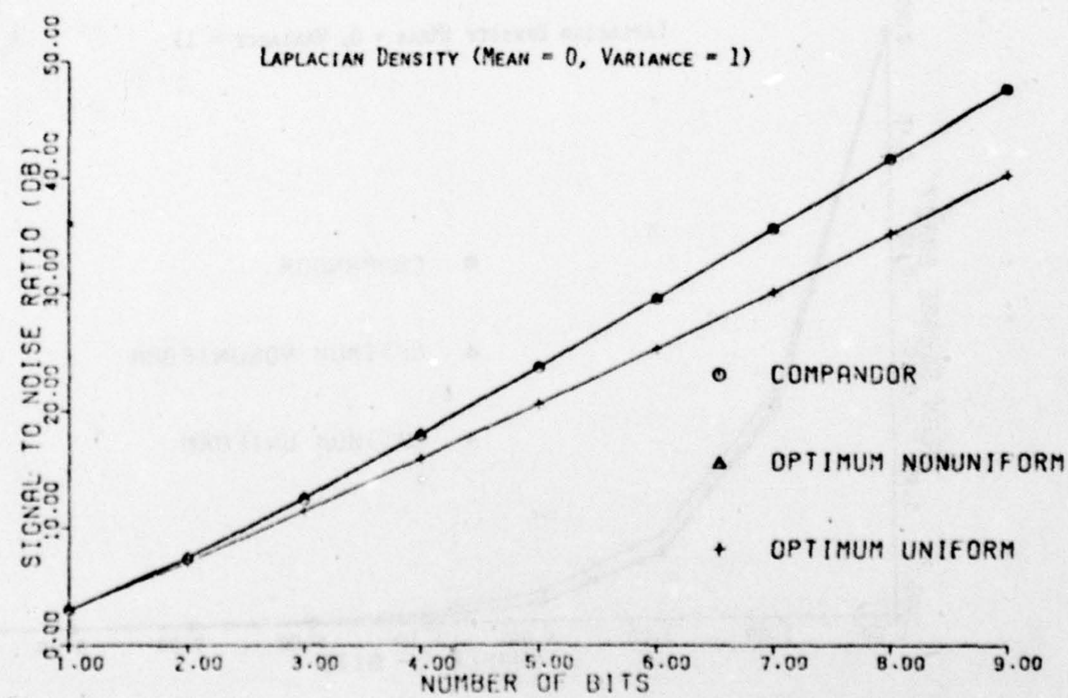


Figure A.8 Signal to Noise Ratio versus Number of Quantizer Bits for Signal with Laplacian Density (Mean = 0, Variance = 1)

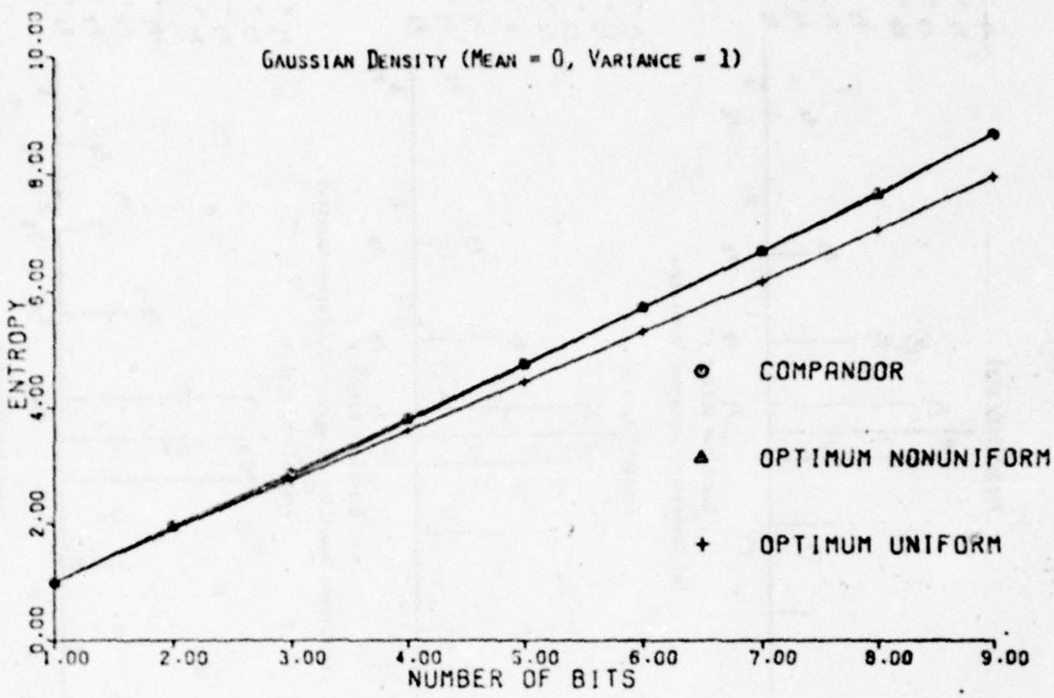


Figure A.9 Entropy versus Number of Quantizer Bits for Signal with Gaussian Density (Mean = 0, Variance = 1)

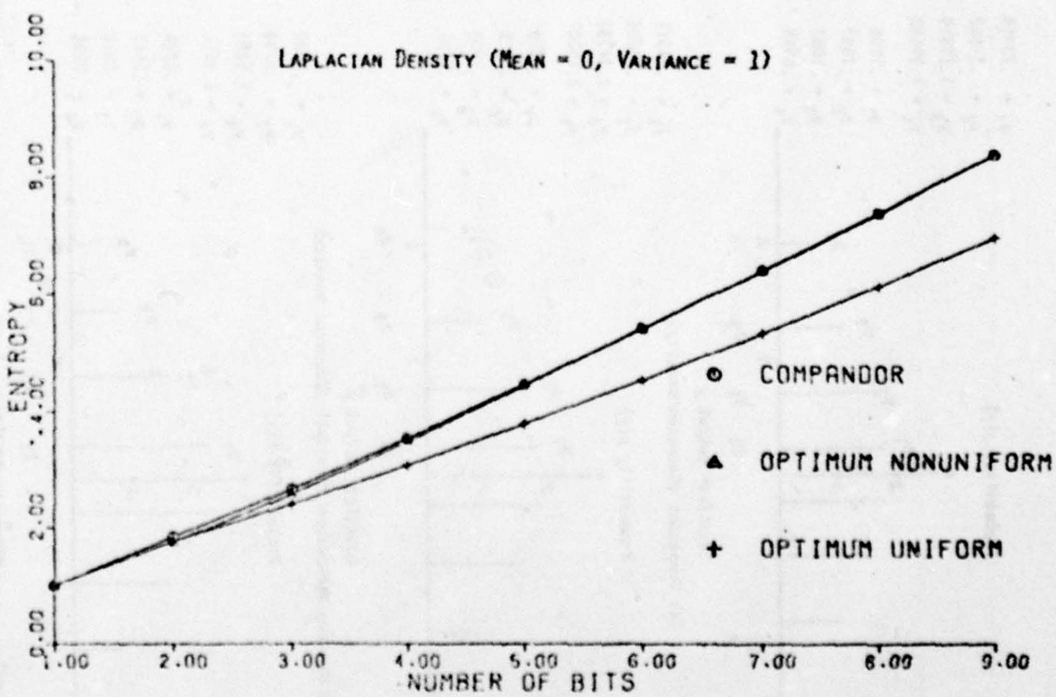
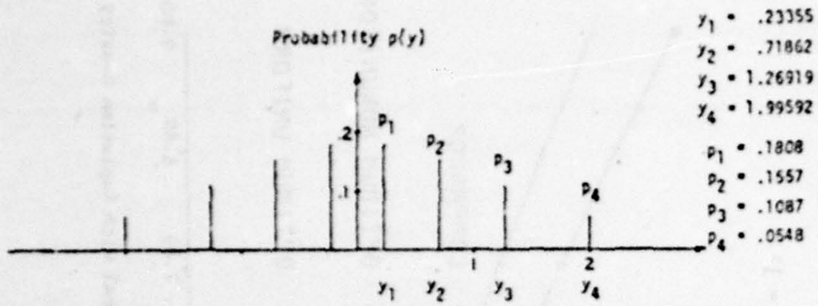
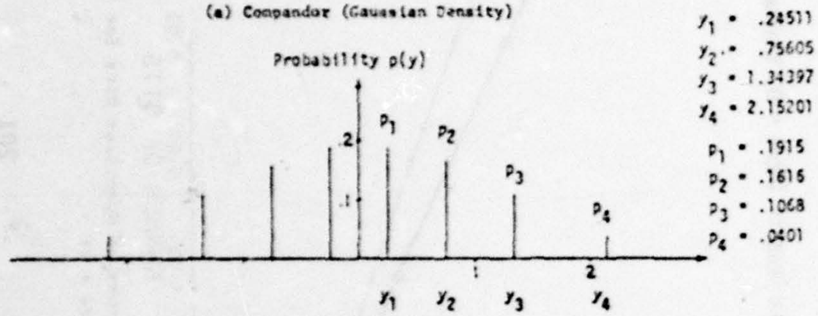


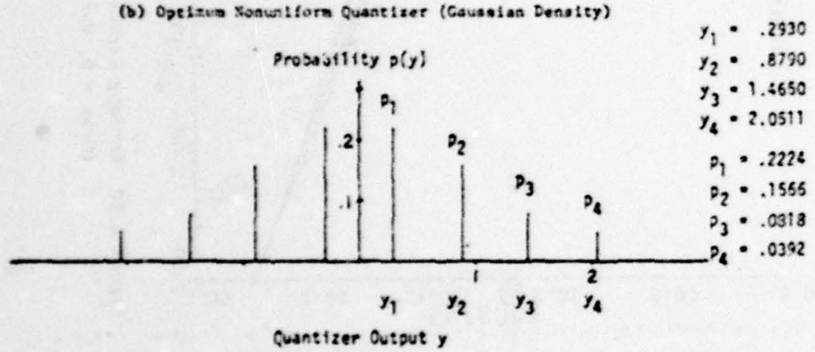
Figure A.10 Entropy versus Number of Quantizer Bits for Signal with Laplacian Density (Mean = 0, Variance = 1)



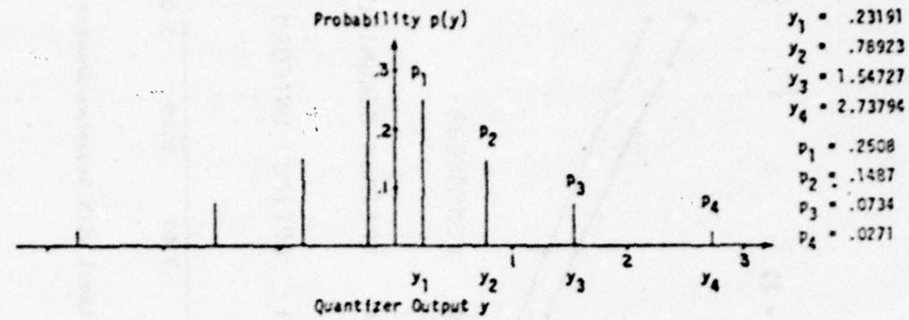
(a) Compandor (Gaussian Density)



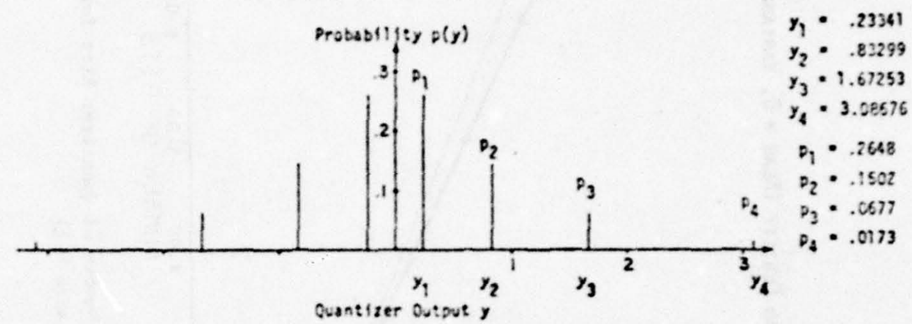
(b) Optimum Nonuniform Quantizer (Gaussian Density)



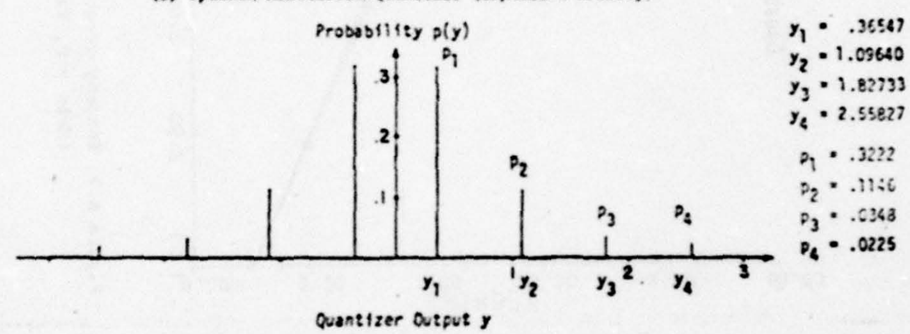
(c) Optimum Uniform Quantizer (Gaussian Density)



(a) Compandor (Laplacian Density)



(b) Optimum Nonuniform Quantizer (Laplacian Density)



(c) Optimum Uniform Quantizer (Laplacian Density)

Figure A.11 Quantizer Output Levels and their Probability Distribution (N = 8)

Table A.1 Approximate Average Amount of CDC Cyber 173 Computer Time Required to Determine the Quantizer Structure with Gaussian Distributed Input Signal

No. of Quantizer Bit n	CDC Cyber 173 Computer Time (Seconds)		
	Optimum Uniform	Optimum Nonuniform	Approximate Nonuniform (Compandor)
1			
2			
3			
4			
5			
6			
7		382.331	
8			
9	10.193		4.816

The times shown are the cumulative computation times that are required to calculate the quantizers up to the corresponding value of n.

Table A.2 Approximate Average Amount of CDC Cyber 173 Computer Time Required to Determine the Quantizer Structure with Laplacian Distributed Input Signal

No. of Quantizer Bit n	CDC Cyber 173 Computer Time (Seconds)		
	Optimum Uniform	Optimum Nonuniform	Approximate Nonuniform (Compandor)
1			
2			
3			
4			
5			
6			
7		128.628	
8			
9	8.504		3.904

The times shown are the cumulative computation times that are required to calculate the quantizers up to the corresponding value of n.

Table A.3 Optimum Uniform Quantizers for Signal with Gaussian Density (Mean = 0, Variance = 1)

NO. OF BITS	NO. OF OUTPUT LEVELS	STEP SIZE	M.S.E.	S.N.R.(DB)	ENTROPY
1	2	1.59577	.36338	4.39639	1.00000
2	4	.99569	.11885	9.25015	1.90373
3	8	.58602	.03744	14.26668	2.76057
4	16	.33520	.01154	19.37686	3.60242
5	32	.18814	.00350	24.56527	4.44926
6	64	.10406	.00104	29.82948	5.30869
7	128	.05687	.00030	35.16651	6.18233
8	256	.03076	.00009	40.57069	7.06944
9	512	.01650	.00002	46.03469	7.96846

Table A.4 Optimum Uniform Quantizers for Signal with Laplacian Density (Mean = 0, Variance = 1)

NO. OF BITS	NO. OF OUTPUT LEVELS	STEP SIZE	M.S.E.	S.N.R.(DB)	ENTROPY
1	2	1.41421	.50000	3.01030	1.00000
2	4	1.05739	.19630	7.07075	1.75066
3	8	.73093	.07175	11.44191	2.39186
4	16	.46100	.02535	15.96005	3.06342
5	32	.27599	.00873	20.59821	3.78130
6	64	.16566	.00291	25.35595	4.53703
7	128	.09610	.00095	30.22918	5.32231
8	256	.05464	.00030	35.20827	6.13130
9	512	.03088	.00009	40.28120	6.95978

Table A.5 Optimum Nonuniform Quantizers for Signal with Gaussian Density (Mean = 0, Variance = 1)

BIT	1		2		3		4		5		6		7	
LEVEL	2		4		8		16		32		64		128	
M.S.E.	.36338		.11748		.03455		.00950		.00250		.00064		.00016	
S.N.R.(DB)	4.39639		9.30029		14.61340		20.22230		26.01244		31.90940		37.86362	
ENTROPY	1.00000		1.91104		2.82443		3.76516		4.72911		5.70744		6.69916	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.79788	0.00000	.45279	0.00000	.24511	0.00000	.12843	0.00000	.06597	0.00000	.03356	0.00000	.01712
2			.98162	1.51043	.50059	.75605	.25830	.38816	.13212	.19828	.06716	.10072	.03425	.05137
3					1.05002	1.34397	.52255	.65693	.26502	.33175	.13438	.16804	.06851	.08564
4					1.74800	2.15201	.79975	.94256	.39948	.46721	.20182	.23560	.10279	.11994
5							1.09953	1.25640	.53639	.60557	.26950	.30351	.13711	.15428
6							1.43741	1.61832	.67674	.74789	.33770	.37188	.17147	.18867
7							1.84381	2.06929	.82166	.89542	.40635	.44082	.20590	.22313
8							2.40106	2.73283	.97258	1.04972	.47564	.51045	.24040	.25767
9									1.13120	1.21281	.54567	.58089	.27499	.29231
10									1.30011	1.38748	.61659	.65228	.30968	.32705
11									1.48236	1.57731	.68853	.72478	.34448	.36190
12									1.68283	1.78832	.76166	.79853	.37940	.39689
13									1.90907	2.02980	.83615	.87376	.41447	.43203
14									2.17430	2.31877	.91218	.95060	.44968	.46733
15									2.50544	2.69280	.98994	1.02935	.48507	.50281
16									2.97686	3.26150	1.06981	1.11025	.52065	.53847
17											1.15194	1.19362	.55642	.57435
18											1.23672	1.27980	.59241	.61046
19											1.32453	1.36924	.62864	.64681
20											1.41585	1.46245	.66512	.68342
21											1.51120	1.56005	.70187	.72031
22											1.61146	1.66286	.73892	.75751
23											1.71733	1.77180	.77628	.79503
24											1.83002	1.88822	.81397	.83290
25											1.95103	2.01382	.85203	.87115
26											2.08240	2.15098	.89048	.90979
27											2.22701	2.30304	.92933	.94886
28											2.38916	2.47529	.96863	.98838
29											2.57367	2.67607	1.00840	1.02840
30											2.79056	2.92182	1.04868	1.06895
31											3.08242	3.24381	1.08949	1.11003
32											3.49567	3.74712	1.13089	1.15173
33													1.17291	1.19407
34													1.21559	1.23709
35													1.25898	1.28086
36													1.30319	1.32542
37													1.34816	1.37084
38													1.39402	1.41718
39													1.44080	1.46452
40													1.48876	1.51294
41													1.53775	1.56253
42													1.58797	1.61340
43													1.63954	1.66565
44													1.69255	1.71943
45													1.74716	1.77489
46													1.80353	1.83216
47													1.86183	1.89144
48													1.92228	1.95307
49													1.98514	2.01719
50													2.05068	2.08410
51													2.11926	2.15435
52													2.19130	2.22843
53													2.26731	2.30637
54													2.34792	2.38965
55													2.43393	2.47840
56													2.52639	2.57436
57													2.62664	2.67841
58													2.73656	2.79420
59													2.85878	2.92335
60													2.99724	3.07111
61													3.15822	3.24530
62													3.35276	3.46020
63													3.60359	3.74696
64													3.97381	4.20005

Table A.5 (cont'd) Optimum Nonuniform Quantizers for Signal with Gaussian Density
(Mean = 0, Variance = 1)

	8	9
BIT	8	9
LEVEL	256	512
M.S.E.	0.00004	0.00001
S.N.R.(DB)	43.82561	49.64151
ENTROPY	7.65260	8.69400

I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)
1	0.000000	.009014	65	1.2218	1.2326	1	0.000000	.0075065	65	.96096	.96845	129	1.9265	1.9330	193	2.9287	2.9371
2	.018029	.027043	66	1.2435	1.2544	2	.015013	.022519	66	.97598	.98347	130	1.9396	1.9472	194	2.9457	2.9542
3	.036058	.045073	67	1.2654	1.2764	3	.030027	.037532	67	.99100	.99849	131	1.9548	1.9623	195	2.9629	2.9716
4	.054091	.063107	68	1.2875	1.2986	4	.045040	.052545	68	1.0060	1.0135	132	1.9699	1.9775	196	2.9801	2.9887
5	.072127	.081145	69	1.3098	1.3209	5	.060053	.067558	69	1.0210	1.0285	133	1.9851	1.9926	197	2.9975	3.0061
6	.090168	.099189	70	1.3322	1.3435	6	.075066	.082572	70	1.0361	1.0436	134	2.0003	2.0078	198	3.0150	3.0237
7	.10822	.11726	71	1.3548	1.3662	7	.090080	.097585	71	1.0511	1.0586	135	2.0154	2.0230	199	3.0325	3.0413
8	.12627	.13530	72	1.3776	1.3891	8	.10509	.11260	72	1.0661	1.0736	136	2.0306	2.0382	200	3.0502	3.0590
9	.14434	.15337	73	1.4007	1.4122	9	.12011	.12761	73	1.0811	1.0886	137	2.0458	2.0534	201	3.0680	3.0769
10	.16241	.17145	74	1.4239	1.4356	10	.13512	.14262	74	1.0962	1.1036	138	2.0610	2.0686	202	3.0859	3.0949
11	.18050	.18954	75	1.4474	1.4591	11	.15013	.15764	75	1.1112	1.1187	139	2.0762	2.0838	203	3.1040	3.1130
12	.19860	.20765	76	1.4711	1.4829	12	.16515	.17265	76	1.1262	1.1337	140	2.0914	2.0990	204	3.1222	3.1313
13	.21671	.22577	77	1.4950	1.5070	13	.18016	.18766	77	1.1412	1.1487	141	2.1067	2.1142	205	3.1406	3.1498
14	.23484	.24390	78	1.5192	1.5313	14	.19517	.20268	78	1.1563	1.1637	142	2.1219	2.1295	206	3.1591	3.1684
15	.25298	.26206	79	1.5436	1.5559	15	.21019	.21769	79	1.1713	1.1788	143	2.1371	2.1447	207	3.1778	3.1872
16	.27115	.28023	80	1.5683	1.5807	16	.22520	.23270	80	1.1863	1.1938	144	2.1524	2.1600	208	3.1967	3.2061
17	.28933	.29843	81	1.5933	1.6059	17	.24021	.24772	81	1.2013	1.2088	145	2.1676	2.1752	209	3.2157	3.2252
18	.30754	.31664	82	1.6186	1.6313	18	.25523	.26273	82	1.2164	1.2239	146	2.1829	2.1905	210	3.2350	3.2446
19	.32577	.33488	83	1.6442	1.6571	19	.27024	.27774	83	1.2314	1.2389	147	2.1982	2.2058	211	3.2544	3.2641
20	.34402	.35315	84	1.6701	1.6831	20	.28525	.29276	84	1.2464	1.2539	148	2.2135	2.2211	212	3.2741	3.2839
21	.36230	.37144	85	1.6964	1.7096	21	.30027	.30777	85	1.2615	1.2689	149	2.2288	2.2364	213	3.2939	3.3039
22	.38060	.38976	86	1.7230	1.7364	22	.31528	.32278	86	1.2765	1.2840	150	2.2441	2.2517	214	3.3141	3.3241
23	.39893	.40810	87	1.7500	1.7636	23	.33030	.33780	87	1.2915	1.2990	151	2.2594	2.2671	215	3.3344	3.3446
24	.41730	.42648	88	1.7774	1.7911	24	.34531	.35281	88	1.3066	1.3140	152	2.2748	2.2824	216	3.3551	3.3654
25	.43569	.44490	89	1.8052	1.8192	25	.36032	.36782	89	1.3216	1.3291	153	2.2902	2.2978	217	3.3760	3.3865
26	.45412	.46336	90	1.8334	1.8476	26	.37534	.38284	90	1.3366	1.3441	154	2.3055	2.3132	218	3.3972	3.4079
27	.47259	.48182	91	1.8621	1.8765	27	.39035	.39785	91	1.3517	1.3591	155	2.3209	2.3286	219	3.4188	3.4296
28	.49109	.50034	92	1.8912	1.9059	28	.40536	.41286	92	1.3667	1.3742	156	2.3363	2.3440	220	3.4407	3.4516
29	.50962	.51890	93	1.9209	1.9358	29	.42038	.42788	93	1.3817	1.3892	157	2.3518	2.3594	221	3.4629	3.4741
30	.52820	.53750	94	1.9511	1.9663	30	.43539	.44289	94	1.3968	1.4043	158	2.3672	2.3749	222	3.4855	3.4969
31	.54682	.55614	95	1.9819	1.9974	31	.45041	.45790	95	1.4118	1.4193	159	2.3827	2.3904	223	3.5086	3.5201
32	.56549	.57483	96	2.0132	2.0290	32	.46542	.47292	96	1.4269	1.4343	160	2.3982	2.4059	224	3.5320	3.5438
33	.58420	.59356	97	2.0452	2.0614	33	.48043	.48793	97	1.4419	1.4494	161	2.4137	2.4214	225	3.5559	3.5680
34	.60296	.61234	98	2.0779	2.0944	34	.49545	.50295	98	1.4569	1.4644	162	2.4292	2.4369	226	3.5804	3.5927
35	.62177	.63118	99	2.1113	2.1281	35	.51046	.51796	99	1.4720	1.4795	163	2.4447	2.4525	227	3.6053	3.6179
36	.64063	.65008	100	2.1454	2.1627	36	.52548	.53298	100	1.4870	1.4945	164	2.4603	2.4681	228	3.6308	3.6437
37	.65954	.66900	101	2.1804	2.1981	37	.54049	.54799	101	1.5021	1.5096	165	2.4759	2.4837	229	3.6570	3.6701
38	.67851	.68800	102	2.2162	2.2343	38	.55551	.56300	102	1.5171	1.5246	166	2.4915	2.4993	230	3.6837	3.6972
39	.69753	.70706	103	2.2530	2.2716	39	.57052	.57802	103	1.5322	1.5397	167	2.5072	2.5150	231	3.7112	3.7251
40	.71662	.72617	104	2.2908	2.3099	40	.58554	.59303	104	1.5472	1.5547	168	2.5229	2.5307	232	3.7395	3.7538
41	.73577	.74536	105	2.3296	2.3493	41	.60055	.60805	105	1.5623	1.5698	169	2.5386	2.5464	233	3.7686	3.7833
42	.75499	.76460	106	2.3696	2.3899	42	.61557	.62306	106	1.5774	1.5849	170	2.5543	2.5621	234	3.7985	3.8137
43	.77427	.78392	107	2.4109	2.4319	43	.63058	.63808	107	1.5924	1.5999	171	2.5701	2.5779	235	3.8295	3.8452
44	.79362	.80331	108	2.4536	2.4753	44	.64560	.65309	108	1.6075	1.6150	172	2.5859	2.5937	236	3.8614	3.8779
45	.81305	.82277	109	2.4978	2.5203	45	.66061	.66811	109	1.6225	1.6300	173	2.6017	2.6096	237	3.8948	3.9116
46	.83255	.84231	110	2.5436	2.5669	46	.67563	.68312	110	1.6376	1.6451	174	2.6176	2.6255	238	3.9293	3.9468
47	.85213	.86193	111	2.5912	2.6155	47	.69064	.69816	111	1.6527	1.6602	175	2.6335	2.6414	239	3.9652	3.9835
48	.87179	.88163	112	2.6409	2.6662	48	.70566	.71315	112	1.6678	1.6753	176	2.6496	2.6574	240	4.0028	4.0219
49	.89153	.90142	113	2.6928	2.7193	49	.72068	.72817	113	1.6828	1.6903	177	2.6654	2.6734	241	4.0421	4.0622
50	.91136	.92129	114	2.7472	2.7750	50	.73569	.74319	114	1.6979	1.7054	178	2.6815	2.6894	242	4.0835	4.1048
51	.93128	.94126	115	2.8044	2.8337	51	.75071	.75820	115	1.7130	1.7205	179	2.6975	2.7055	243	4.1271	4.1494
52	.95130	.96132	116	2.8648	2.8958	52	.76573	.77322	116	1.7281	1.7356	180	2.7137	2.7217	244	4.1733	4.1970
53	.97141	.98148	117	2.9288	2.9618	53	.78075	.78824	117	1.7432	1.7507	181	2.7298	2.7379	245	4.2225	4.2478
54	.99162	1.0017	118	2.9971	3.0324	54	.79576	.80325	118	1.7582	1.7658	182	2.7461	2.7541	246	4.2751	4.3023
55	1.0119	1.0221	119	3.0704	3.1084	55	.81078	.81827	119	1.7733	1.7809	183	2.7623	2.7705	247	4.3319	4.3613
56	1.0324	1.0426	120	3.1497	3.1909	56	.82579	.83329	120	1.7884	1.7959	184	2.7787	2.7868	248	4.3935	4.4256
57	1.0529	1.0632	121	3.2361	3.2813	57	.84081	.84830	121	1.8035	1.8111	185	2.7951	2.8032	249	4.4611	4.4964
58	1.0736	1.0839	122	3.3314	3.3815	58	.85583	.86332	122	1.8186	1.8262	186	2.8115	2.8197	250	4.5360	4.5755
59	1.0943	1.1047	123	3.4380	3.4945	59	.87085	.87834	123	1.8337	1.8413	187	2.8280	2.8363	251	4.6206	4.6653
60	1.1152	1.1257	124	3.5596	3.6247	60	.88587	.89336	124	1.8489	1.8564	188	2.8446	2.8529	252	4.7174	4.7694
61	1.1363	1.1468	125	3.7020	3.7792	61	.90089	.90839	125	1.8640	1.8715	189	2.8613	2.8696	253	4.8318	4.8942
62	1.1574	1.1680	126	3.8751	3.9714	62	.91590	.92339	126	1.8791	1.8866	190	2.8780	2.8864	254	4.9725	5.0508
63	1.1787	1.1894	127	4.1007	4.2301	63	.93092	.93841	127	1.8942	1.9018	191	2.8948	2.9032	255	5.1575	5.2642
64	1.2002	1.2109	128	4.4369	4.6437	64	.94594	.95343	128	1.9094	1.9169	192	2.9117	2.9201	256	5.4376	5.6166

Table A.6 Optimum Nonuniform Quantizers for Signal with Laplacian Density (Mean = 0, Variance = 1)

BIT	1	2	3	4	5	6	7
LEVEL	2	4	8	16	32	64	128
P.S.C.	.50000	.17619	.05448	.01537	.00410	.00106	.00027
S.N.R.(DB)	3.01030	7.54007	12.63795	18.13295	23.86905	29.74301	35.68799
ENTROPY	1.00000	1.72826	2.56537	3.47471	4.42630	5.40032	6.38259

I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.000000	.707107	0.000000	.419762	0.000000	.233419	0.000000	.124093	0.000000	.064082	0.000000	.032630	0.000000	.016558
2			1.12649	1.83395	.53321	.83299	.26445	.40490	.13229	.20049	.06631	.09999	.03338	.05026
3					1.25277	1.67253	.56663	.72676	.27330	.34627	.13477	.16954	.06729	.08436
4					2.37965	3.08676	.91995	1.11116	.42454	.50281	.20550	.24145	.10175	.11912
5							1.34457	1.57798	.58731	.67181	.27867	.31508	.13678	.15443
6							1.87779	2.17758	.76363	.85505	.35445	.39301	.17239	.19034
7							2.59736	3.01713	.95597	1.05640	.43303	.47304	.20861	.22687
8							3.72425	4.43136	1.16753	1.27856	.51462	.55628	.24545	.26403
9									1.40268	1.52662	.59947	.64279	.28294	.30185
10									1.66710	1.80756	.68775	.73295	.32110	.34035
11									1.96951	2.13144	.78006	.82716	.35995	.37955
12									2.32265	2.51304	.87644	.92572	.39952	.41948
13									2.74720	2.94870	.97740	1.02907	.43983	.46018
14									3.28851	3.56031	1.08339	1.13769	.48092	.50166
15									4.00089	4.41985	1.19493	1.25216	.52281	.54395
16									5.12698	5.83400	1.31265	1.37313	.56553	.58710
17											1.43727	1.50140	.60912	.63113
18											1.56966	1.63790	.65361	.67608
19											1.71083	1.78376	.69934	.72199
20											1.86207	1.94036	.74545	.76898
21											2.02490	2.10942	.79288	.81686
22											2.20127	2.29310	.84139	.86591
23											2.39364	2.49417	.89181	.91610
24											2.60523	2.71627	.94181	.96750
25											2.84032	2.96035	.99383	1.02015
26											3.10404	3.24530	1.04714	1.07412
27											3.40726	3.56920	1.10181	1.12948
28											3.76041	3.95160	1.15790	1.18631
29											4.18504	4.41846	1.21550	1.24467
30											4.71028	5.01007	1.27468	1.30467
31											5.43785	5.85762	1.33553	1.36638
32											6.56474	7.27185	1.39416	1.42993
33													1.46267	1.49540
34													1.52916	1.56294
35													1.59781	1.63267
36													1.66872	1.70475
37													1.74204	1.77933
38													1.81797	1.85668
39													1.89668	1.93675
40													1.97640	2.02003
41													2.06336	2.10668
42													2.15184	2.19698
43													2.24416	2.29127
44													2.34060	2.38993
45													2.44163	2.49333
46													2.54768	2.60281
47													2.65920	2.71653
48													2.77705	2.83755
49													2.90171	2.96585
50													3.03412	3.10238
51													3.17533	3.24826
52													3.32658	3.40489
53													3.48943	3.57396
54													3.66502	3.75765
55													3.85820	3.95873
56													4.06980	4.18085
57													4.30490	4.42893
58													4.56942	4.70908
59													4.87184	5.03378
60													5.22580	5.41619
61													5.64963	5.88305
62													6.18787	6.42264
63													6.90744	7.32220
64													8.02933	8.72644

Table A.6 (cont'd) Optimum Nonuniform Quantizer for Signal with Laplacian Density
(Mean = 0, Variance = 1)

	8	9
BIT	8	9
LEVEL	256	512
M.S.E.	0.0007	.00002
S.N.R. (DB)	41.66436	47.65760
ENTROPY	7.35616	8.38272

I	X(I)	Y(I)	I	X(I)	Y(I)
1	0.000000	.0085027	65	1.5027	1.5195
2	.017074	.025645	66	1.5365	1.5535
3	.034286	.042927	67	1.5708	1.5880
4	.051639	.060350	68	1.6056	1.6341
5	.069133	.077915	69	1.6410	1.6588
6	.086772	.095625	70	1.6769	1.6950
7	.10456	.11369	71	1.7135	1.7319
8	.12269	.13169	72	1.7506	1.7694
9	.14058	.14965	73	1.7884	1.8075
10	.15581	.16797	74	1.8269	1.8463
11	.17720	.18644	75	1.8660	1.8857
12	.19575	.20507	76	1.9058	1.9259
13	.21446	.22386	77	1.9464	1.9669
14	.23336	.24281	78	1.9878	2.0086
15	.25238	.26193	79	2.0299	2.0511
16	.27158	.28122	80	2.0728	2.0945
17	.29096	.30069	81	2.1166	2.1387
18	.31051	.32032	82	2.1613	2.1839
19	.33023	.34014	83	2.2069	2.2300
20	.35013	.36013	84	2.2535	2.2771
21	.37022	.38031	85	2.3011	2.3252
22	.39049	.40067	86	2.3498	2.3744
23	.41095	.42122	87	2.3996	2.4247
24	.43160	.44197	88	2.4505	2.4763
25	.45244	.45291	89	2.5027	2.5291
26	.47349	.48406	90	2.5561	2.5832
27	.49473	.50541	91	2.6109	2.6386
28	.51619	.52696	92	2.6671	2.6956
29	.53785	.54873	93	2.7248	2.7540
30	.55972	.57071	94	2.7841	2.8141
31	.58181	.59291	95	2.8450	2.8759
32	.60412	.61533	96	2.9077	2.9395
33	.62666	.63799	97	2.9723	3.0051
34	.64943	.66087	98	3.0389	3.0727
35	.67243	.68399	99	3.1076	3.1425
36	.69568	.70736	100	3.1786	3.2146
37	.71916	.73097	101	3.2519	3.2892
38	.74290	.75483	102	3.3279	3.3666
39	.76689	.77893	103	3.4067	3.4469
40	.79114	.80333	104	3.4884	3.5301
41	.81566	.82799	105	3.5736	3.6168
42	.84045	.85291	106	3.6619	3.7071
43	.86552	.87812	107	3.7563	3.8014
44	.89087	.90361	108	3.8507	3.9001
45	.91651	.92940	109	3.9518	4.0035
46	.94244	.95568	110	4.0579	4.1122
47	.96868	.98188	111	4.1695	4.2267
48	.99524	1.0086	112	4.2873	4.3478
49	1.0221	1.0356	113	4.4119	4.4761
50	1.0493	1.0630	114	4.5444	4.6126
51	1.0768	1.0907	115	4.6856	4.7585
52	1.1047	1.1187	116	4.8368	4.9151
53	1.1329	1.1471	117	4.9997	5.0842
54	1.1615	1.1759	118	5.1761	5.2679
55	1.1905	1.2050	119	5.3685	5.4690
56	1.2198	1.2346	120	5.5801	5.6911
57	1.2495	1.2645	121	5.8152	5.9392
58	1.2796	1.2948	122	6.0797	6.2201
59	1.3102	1.3255	123	6.3821	6.5460
60	1.3411	1.3567	124	6.7351	6.9265
61	1.3725	1.3883	125	7.1599	7.3933
62	1.4044	1.4206	126	7.6931	7.9929
63	1.4367	1.4529	127	8.4127	8.8325
64	1.4696	1.4859	128	9.5396	10.247

I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)
1	0.000000	.0059117	65	.88470	.89662	129	2.0368	2.0469	193	3.6705	3.6977
2	.011857	.017501	66	.90461	.91256	130	2.0572	2.0674	194	3.7052	3.7226
3	.023779	.029756	67	.92058	.92857	131	2.0777	2.0879	195	3.7403	3.7580
4	.035767	.041777	68	.93662	.94463	132	2.0982	2.1085	196	3.7760	3.7939
5	.047821	.053865	69	.95271	.96076	133	2.1189	2.1293	197	3.8122	3.8304
6	.059941	.066016	70	.96886	.97694	134	2.1397	2.1501	198	3.8489	3.8676
7	.072127	.078235	71	.98508	.99319	135	2.1606	2.1711	199	3.8862	3.9050
8	.084378	.090519	72	1.0014	1.0095	136	2.1816	2.1921	200	3.9241	3.9432
9	.096695	.10287	73	1.0177	1.0259	137	2.2027	2.2133	201	3.9626	3.9820
10	.10908	.11528	74	1.0341	1.0423	138	2.2240	2.2346	202	4.0017	4.0216
11	.12153	.12776	75	1.0505	1.0588	139	2.2453	2.2560	203	4.0415	4.0616
12	.13404	.14031	76	1.0671	1.0753	140	2.2667	2.2775	204	4.0820	4.1024
13	.14662	.15292	77	1.0836	1.0919	141	2.2883	2.2991	205	4.1232	4.1439
14	.15926	.16560	78	1.1003	1.1086	142	2.3100	2.3209	206	4.1641	4.1863
15	.17197	.17834	79	1.1170	1.1253	143	2.3318	2.3427	207	4.2078	4.2293
16	.18475	.19115	80	1.1337	1.1421	144	2.3538	2.3647	208	4.2513	4.2732
17	.19759	.20402	81	1.1506	1.1590	145	2.3758	2.3869	209	4.2956	4.3180
18	.21050	.21696	82	1.1674	1.1759	146	2.3980	2.4091	210	4.3408	4.3637
19	.22347	.22997	83	1.1844	1.1929	147	2.4204	2.4315	211	4.3870	4.4102
20	.23650	.24303	84	1.2014	1.2099	148	2.4428	2.4541	212	4.4340	4.4578
21	.24960	.25617	85	1.2185	1.2270	149	2.4654	2.4768	213	4.4821	4.5066
22	.26277	.26936	86	1.2356	1.2441	150	2.4882	2.4996	214	4.5312	4.5560
23	.27600	.28263	87	1.2528	1.2614	151	2.5111	2.5226	215	4.5816	4.6069
24	.28529	.29595	88	1.2700	1.2787	152	2.5342	2.5457	216	4.6327	4.6587
25	.30265	.30935	89	1.2873	1.2960	153	2.5574	2.5690	217	4.6853	4.7128
26	.31608	.32280	90	1.3047	1.3134	154	2.5807	2.5924	218	4.7391	4.7663
27	.32957	.33632	91	1.3222	1.3309	155	2.6042	2.6160	219	4.7962	4.8271
28	.34312	.34991	92	1.3397	1.3484	156	2.6279	2.6398	220	4.8509	4.8796
29	.35674	.36355	93	1.3572	1.3660	157	2.6519	2.6637	221	4.9088	4.9382
30	.37042	.37727	94	1.3749	1.3837	158	2.6758	2.6878	222	4.9686	4.9986
31	.38416	.39106	95	1.3926	1.4014	159	2.7000	2.7121	223	5.0296	5.0607
32	.39797	.40488	96	1.4103	1.4192	160	2.7244	2.7366	224	5.0926	5.1245
33	.41184	.41879	97	1.4282	1.4371	161	2.7490	2.7613	225	5.1575	5.1904
34	.42578	.43275	98	1.4460	1.4550	162	2.7737	2.7861	226	5.2243	5.2582
35	.43978	.44678	99	1.4640	1.4730	163	2.7987	2.8112	227	5.2932	5.3282
36	.45384	.46088	100	1.4820	1.4910	164	2.8238	2.8364	228	5.3644	5.4006
37	.46796	.47503	101	1.5001	1.5092	165	2.8492	2.8619	229	5.4380	5.4755
38	.48215	.48925	102	1.5183	1.5274	166	2.8748	2.8876	230	5.5142	5.5530
39	.49640	.50354	103	1.5365	1.5456	167	2.9005	2.9135	231	5.5932	5.6336
40	.51072	.51788	104	1.5548	1.5639	168	2.9265	2.9396	232	5.6751	5.7169
41	.52510	.53229	105	1.5732	1.5823	169	2.9528	2.9659	233	5.7603	5.8037
42	.53954	.54676	106	1.5916	1.6008	170	2.9792	2.9925	234	5.8490	5.8942
43	.55404	.56130	107	1.6101	1.6193	171	3.0060	3.0193	235	5.9415	5.9887
44	.56860	.57589	108	1.6287	1.6380	172	3.0329	3.0464	236	6.0381	6.0875
45	.58323	.59055	109	1.6473	1.6566	173	3.0601	3.0738	237	6.1393	6.1911
46	.59792	.60527	110	1.6660	1.6754	174	3.0876	3.1014	238	6.2455	6.2999
47	.61267	.62005	111	1.6848	1.6942	175	3.1153	3.1292	239	6.3572	6.4146
48	.62748	.63490	112	1.7037	1.7131	176	3.1433	3.1574	240	6.4751	6.5357
49	.64236	.64980	113	1.7226	1.7321	177	3.1716	3.1858	241	6.5999	6.6641
50	.65730	.66477	114	1.7417	1.7512	178	3.2002	3.2145	242	6.7325	6.8008
51	.67230	.67980	115	1.7608	1.7703	179	3.2291	3.2436	243	6.8738	6.9467
52	.68736	.69489	116	1.7799	1.7895	180	3.2583	3.2729	244	7.0251	7.1035
53	.70248	.71004	117	1.7992	1.8089	181	3.2878	3.3026	245	7.1880	7.2726
54	.71766	.72525	118	1.8185	1.8282	182	3.3176	3.3326	246	7.3645	7.4566
55	.73290	.74053	119	1.8379	1.8476	183	3.3478	3.3629	247	7.5570	7.6575
56	.74821	.75586	120	1.8574	1.8672	184	3.3783	3.3935	248	7.7686	7.8797
57	.76358	.77126	121	1.8770	1.8868	185	3.4092	3.4247	249	8.0036	8.1278
58	.77900	.78672	122	1.8967	1.9065	186	3.4404	3.4561	250	8.2661	8.4098
59	.79449	.80224	123	1.9164	1.9263	187	3.4771	3.4880	251	8.5706	8.7328
60	.81004	.81782	124	1.9362	1.9462	188	3.5041	3.5202	252	8.9260	9.1152
61	.82565	.83346	125	1.9562	1.9661	189	3.5365	3.5528	253	9.3487	9.5821
62	.84132	.84916	126	1.9762	1.9862	190	3.5693	3.5859	254	9.8819	10.182
63	.85705	.86492	127	1.9963	2.0063	191	3.6026	3.6193	255	10.602	11.021
64	.87284	.88074	128	2.0165	2.0266	192	3.6363	3.6533	256	11.728	12.436

Table A.7 Compandors for Signal with Gaussian Density
(Mean = 0, Variance = 1)

BIT	1	2	3	4	5	6	7
LEVEL	2	4	8	16	32	64	128
M.S.E.	.36338	.12035	.03564	.00988	.00257	.00066	.00017
S.N.R.(DB)	4.39707	9.19543	14.48067	20.05016	25.90319	31.83331	37.61614
ENTROPY	1.00000	1.99876	2.88304	3.81166	4.76013	5.72790	6.70939

I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.79788	0.00000	.42426	0.00000	.23355	0.00000	.12471	0.00000	.06453	0.00000	.03301	0.00000	.01672
2	1.62146		.97607	1.29829	.47144	.71862	.24967	.37524	.12916	.19396	.06602	.09907	.03343	.05015
3			2.05788		.98145	1.26919	.50343	.63445	.25903	.32448	.13214	.16527	.06688	.08361
4					1.55725	1.99592	.76920	.90879	.39039	.45687	.19846	.23172	.10035	.11710
5					2.54383		1.05455	1.20828	.52403	.59199	.26506	.29851	.13386	.15063
6							1.37201	1.54912	.66287	.73081	.33206	.36574	.16741	.18421
7							1.74414	1.96424	.80197	.87451	.39956	.43354	.20103	.21787
8							2.22169	2.54058	.94861	1.02450	.46760	.50200	.23473	.25161
9							2.98880		1.10240	1.18260	.53652	.57126	.26852	.28545
10									1.26543	1.35125	.60622	.64144	.30240	.31939
11									1.44052	1.53380	.67692	.71269	.33641	.35346
12									1.63176	1.73523	.74877	.78517	.37054	.38766
13									1.84530	1.96337	.82193	.85906	.40482	.42202
14									2.09136	2.23191	.89656	.93456	.43926	.45654
15									2.38892	2.56847	.97294	1.01184	.47387	.49125
16									2.78085	3.04586	1.05124	1.09120	.50865	.52615
17									3.41066		1.13175	1.17294	.54369	.56127
18											1.21479	1.25737	.57892	.59663
19											1.30071	1.34489	.61429	.63223
20											1.38997	1.43600	.65013	.66810
21											1.48567	1.53127	.68614	.70426
22											1.58068	1.63141	.72245	.74072
23											1.68358	1.73732	.75909	.77753
24											1.79279	1.85017	.79606	.81465
25											1.90965	1.97147	.83340	.85222
26											2.03591	2.10336	.87114	.89016
27											2.17403	2.24858	.90929	.92853
28											2.32756	2.41169	.94789	.96736
29											2.50193	2.59953	.98696	1.00669
30											2.70616	2.82416	1.02654	1.04654
31											2.95694	3.10988	1.06667	1.08695
32											3.29168	3.51915	1.10738	1.12796
33											3.83115		1.14871	1.16962
34													1.19070	1.21196
35													1.23341	1.25504
36													1.27688	1.29891
37													1.32116	1.34363
38													1.36633	1.38927
39													1.41245	1.43589
40													1.45959	1.48358
41													1.50784	1.53241
42													1.55729	1.58250
43													1.60804	1.63394
44													1.66022	1.68687
45													1.71390	1.74143
46													1.76936	1.79777
47													1.82666	1.85607
48													1.88603	1.91657
49													1.94771	1.97949
50													2.01195	2.04514
51													2.07939	2.11387
52													2.14952	2.18610
53													2.22360	2.26234
54													2.30216	2.34324
55													2.38568	2.42960
56													2.47514	2.52247
57													2.57175	2.62321
58													2.67709	2.73370
59													2.79338	2.85654
60													2.92383	2.99560
61													3.07335	3.15761
62													3.25007	3.35284
63													3.46899	3.60321
64													3.76337	3.96419
65													4.23070	

Table A.7 (cont'd) Compandors for Signal with Gaussian Density (Mean = 0, Variance = 1)

BIT	8	9
LEVEL	256	512
M.S.E.	0.00006	0.00001
S.N.R. (DB)	43.82561	49.84151
ENTROPY	7.96631	8.69502

I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)
1	0.000000	.0084161	65	1.1530	1.1580	1	0.0000000	.0042235	65	.54970	.55414	129	1.1630	1.1683	193	1.9803	1.9885
2	.016832	.025249	66	1.1792	1.1895	2	.0084620	.012671	66	.55859	.56304	130	1.1736	1.1789	194	1.9967	2.0049
3	.033666	.042085	67	1.2005	1.2112	3	.016894	.021118	67	.56749	.57195	131	1.1843	1.1896	195	2.0132	2.0215
4	.050503	.058926	68	1.2220	1.2328	4	.025342	.029566	68	.57641	.58088	132	1.1949	1.2003	196	2.0299	2.0383
5	.067345	.075769	69	1.2437	1.2546	5	.033790	.038015	69	.58535	.58982	133	1.2057	1.2111	197	2.0468	2.0553
6	.084193	.092620	70	1.2655	1.2766	6	.042239	.046464	70	.59430	.59878	134	1.2165	1.2219	198	2.0638	2.0724
7	.10105	.10948	71	1.2876	1.2988	7	.050659	.054915	71	.60327	.60776	135	1.2273	1.2327	199	2.0811	2.0898
8	.11792	.12635	72	1.3099	1.3212	8	.059141	.063367	72	.61225	.61675	136	1.2382	1.2436	200	2.0986	2.1074
9	.13479	.14324	73	1.3326	1.3438	9	.067593	.071820	73	.62125	.62576	137	1.2491	1.2546	201	2.1163	2.1253
10	.15168	.16013	74	1.3552	1.3666	10	.076047	.080275	74	.63027	.63478	138	1.2601	1.2655	202	2.1342	2.1433
11	.16859	.17706	75	1.3782	1.3897	11	.084504	.088732	75	.63930	.64383	139	1.2711	1.2767	203	2.1524	2.1616
12	.18551	.19397	76	1.4014	1.4131	12	.092962	.097192	76	.64835	.65289	140	1.2822	1.2878	204	2.1708	2.1801
13	.20244	.21092	77	1.4249	1.4367	13	.10142	.10565	77	.65742	.66196	141	1.2934	1.2989	205	2.1895	2.1989
14	.21940	.22789	78	1.4486	1.4606	14	.10988	.11412	78	.66651	.67106	142	1.3045	1.3102	206	2.2084	2.2179
15	.23638	.24488	79	1.4726	1.4847	15	.11835	.12258	79	.67562	.68017	143	1.3158	1.3214	207	2.2276	2.2372
16	.25338	.26189	80	1.4969	1.5092	16	.12682	.13105	80	.68474	.68931	144	1.3271	1.3328	208	2.2470	2.2566
17	.27041	.27893	81	1.5215	1.5340	17	.13529	.13953	81	.69388	.69846	145	1.3384	1.3441	209	2.2668	2.2767
18	.28746	.29600	82	1.5465	1.5590	18	.14376	.14800	82	.70304	.70763	146	1.3499	1.3556	210	2.2868	2.2969
19	.30454	.31309	83	1.5717	1.5844	19	.15224	.15648	83	.71221	.71683	147	1.3613	1.3671	211	2.3071	2.3174
20	.32165	.33021	84	1.5973	1.6102	20	.16072	.16497	84	.72143	.72604	148	1.3729	1.3787	212	2.3278	2.3383
21	.33879	.34737	85	1.6232	1.6363	21	.16921	.17345	85	.73065	.73527	149	1.3845	1.3903	213	2.3488	2.3595
22	.35596	.36456	86	1.6495	1.6628	22	.17770	.18194	86	.73990	.74453	150	1.3961	1.4020	214	2.3702	2.3810
23	.37317	.38179	87	1.6762	1.6897	23	.18619	.19044	87	.74916	.75380	151	1.4078	1.4137	215	2.3919	2.4029
24	.39042	.39906	88	1.7033	1.7170	24	.19469	.19894	88	.75845	.76310	152	1.4196	1.4255	216	2.4140	2.4232
25	.40770	.41636	89	1.7308	1.7447	25	.20319	.20745	89	.76775	.77242	153	1.4315	1.4374	217	2.4366	2.4480
26	.42503	.43371	90	1.7587	1.7729	26	.21170	.21596	90	.77708	.78176	154	1.4434	1.4494	218	2.4595	2.4731
27	.44240	.45110	91	1.7872	1.8016	27	.22021	.22447	91	.78644	.79112	155	1.4554	1.4614	219	2.4829	2.4967
28	.45981	.46853	92	1.8161	1.8307	28	.22873	.23299	92	.79581	.80051	156	1.4674	1.4735	220	2.5067	2.5188
29	.47727	.48602	93	1.8455	1.8605	29	.23726	.24152	93	.80521	.80992	157	1.4796	1.4857	221	2.5310	2.5434
30	.49478	.50355	94	1.8755	1.8907	30	.24579	.25005	94	.81463	.81935	158	1.4918	1.4979	222	2.5558	2.5685
31	.51234	.52114	95	1.9060	1.9215	31	.25432	.25859	95	.82408	.82881	159	1.5041	1.5102	223	2.5812	2.5941
32	.52995	.53877	96	1.9372	1.9530	32	.26286	.26714	96	.83355	.83830	160	1.5164	1.5226	224	2.6071	2.6203
33	.54761	.55647	97	1.9690	1.9851	33	.27141	.27569	97	.84305	.84781	161	1.5288	1.5351	225	2.6337	2.6472
34	.56534	.57422	98	2.0014	2.0179	34	.27997	.28425	98	.85257	.85734	162	1.5414	1.5476	226	2.6608	2.6747
35	.58312	.59204	99	2.0346	2.0515	35	.28853	.29281	99	.86212	.86690	163	1.5539	1.5603	227	2.6887	2.7024
36	.60097	.60991	100	2.0685	2.0858	36	.29710	.30138	100	.87169	.87649	164	1.5666	1.5730	228	2.7172	2.7318
37	.61887	.62785	101	2.1033	2.1210	37	.30567	.30996	101	.88130	.88611	165	1.5794	1.5858	229	2.7465	2.7614
38	.63685	.64586	102	2.1389	2.1571	38	.31425	.31855	102	.89092	.89575	166	1.5922	1.5987	230	2.7766	2.7920
39	.65489	.66394	103	2.1755	2.1941	39	.32285	.32714	103	.90058	.90542	167	1.6052	1.6117	231	2.8076	2.8234
40	.67301	.68209	104	2.2130	2.2322	40	.33144	.33575	104	.91026	.91511	168	1.6182	1.6247	232	2.8395	2.8558
41	.69120	.70032	105	2.2517	2.2714	41	.34005	.34436	105	.91997	.92484	169	1.6313	1.6379	233	2.8724	2.8892
42	.70946	.71863	106	2.2916	2.3118	42	.34867	.35298	106	.92972	.93460	170	1.6445	1.6511	234	2.9063	2.9237
43	.72781	.73701	107	2.3325	2.3535	43	.35729	.36161	107	.93949	.94438	171	1.6578	1.6645	235	2.9414	2.9595
44	.74624	.75548	108	2.3748	2.3964	44	.36593	.37025	108	.94929	.95420	172	1.6712	1.6780	236	2.9778	2.9965
45	.76475	.77404	109	2.4187	2.4412	45	.37457	.37889	109	.95912	.96405	173	1.6847	1.6915	237	3.0155	3.0350
46	.78335	.79269	110	2.4642	2.4873	46	.38322	.38755	110	.96896	.97392	174	1.6983	1.7052	238	3.0548	3.0750
47	.80204	.81142	111	2.5114	2.5357	47	.39188	.39621	111	.97887	.98383	175	1.7120	1.7189	239	3.0956	3.1167
48	.82082	.83025	112	2.5605	2.5859	48	.40055	.40489	112	.98880	.99378	176	1.7259	1.7328	240	3.1381	3.1603
49	.83970	.84918	113	2.6115	2.6385	49	.40923	.41358	113	.99876	1.0038	177	1.7398	1.7468	241	3.1829	3.2060
50	.85869	.86822	114	2.6655	2.6934	50	.41792	.42227	114	1.0088	1.0138	178	1.7538	1.7609	242	3.2299	3.2541
51	.87777	.88735	115	2.7219	2.7513	51	.42663	.43098	115	1.0188	1.0238	179	1.7680	1.7751	243	3.2791	3.3044
52	.89696	.90660	116	2.7814	2.8124	52	.43534	.43970	116	1.0288	1.0339	180	1.7823	1.7895	244	3.3313	3.3585
53	.91627	.92596	117	2.8443	2.8727	53	.44406	.44843	117	1.0389	1.0440	181	1.7967	1.8039	245	3.3866	3.4157
54	.93569	.94544	118	2.9112	2.9444	54	.45280	.45717	118	1.0491	1.0541	182	1.8112	1.8185	246	3.4457	3.4768
55	.95522	.96506	119	2.9828	3.0206	55	.46155	.46592	119	1.0592	1.0643	183	1.8259	1.8333	247	3.5091	3.5426
56	.97498	.98476	120	3.0599	3.1004	56	.47030	.47468	120	1.0694	1.0745	184	1.8407	1.8481	248	3.5775	3.6160
57	.99467	1.0056	121	3.1435	3.1892	57	.47907	.48346	121	1.0797	1.0848	185	1.8556	1.8631	249	3.6521	3.6920
58	1.0146	1.0246	122	3.2352	3.2856	58	.48785	.49225	122	1.0899	1.0951	186	1.8707	1.8782	250	3.7341	3.7786
59	1.0346	1.0447	123	3.3369	3.3925	59	.49665	.50105	123	1.1003	1.1054	187	1.8859	1.8935	251	3.8236	3.8754
60	1.0548	1.0650	124	3.4517	3.5153	60	.50546	.50987	124	1.1106	1.1158	188	1.9012	1.9090	252	3.9238	3.9862
61	1.0752	1.0854	125	3.5840	3.6589	61	.51428	.51869	125	1.1210	1.1262	189	1.9167	1.9245	253	4.0484	4.1161
62	1.0956	1.1059	126	3.7413	3.8331	62	.52313	.52753	126	1.1314	1.1367	190	1.9324	1.9403	254	4.1912	4.2748
63	1.1163	1.1267	127	3.9372	4.0573	63	.53196	.53639	127	1.1419	1.1472	191	1.9482	1.9562	255	4.3698	4.4801
64	1.1371	1.1475	128	4.2021	4.3832	64	.54082	.54526	128	1.1524	1.1577	192	1.9642	1.9722	256	4.6123	4.7785
			129	4.6305											257	5.0051	

Table A.8 Compandors for Signal with Laplacian Density
(Mean = 0, Variance = 1)

BIT	1	2	3	4	5	6	7
LEVEL	2	4	8	16	32	64	128
P.S.E.	.50038	.18346	.05714	.01603	.00423	.00108	.00027
S.N.R.(DB)	3.00962	7.34461	12.43090	17.95036	23.73534	29.65610	35.63736
ENTROPY	1.00000	1.83010	2.65404	3.53310	4.46120	5.42113	6.39967

I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.69827	0.00000	.42089	0.00000	.23191	0.00000	.12781	0.00000	.06354	0.00000	.03240	0.00000	.01638
2	1.74617		.94648	1.44677	.49232	.76923	.25318	.39205	.12905	.19664	.06531	.09873	.03286	.04952
3			2.69924		1.13457	1.54727	.54271	.70056	.26646	.33865	.13269	.16720	.06628	.08318
4					2.06014	2.73794	.87543	1.06165	.41339	.49065	.20228	.23795	.10022	.11739
5					3.73994		1.26022	1.49709	.57126	.65463	.27423	.31114	.13470	.15216
6							1.75367	2.04562	.74183	.83255	.34871	.38695	.16976	.18750
7							2.38425	2.76741	.92732	1.02653	.42596	.46557	.20540	.22345
8							3.27560	3.93790	1.13061	1.24205	.50622	.54722	.24166	.26002
9							4.88469		1.35545	1.47749	.58925	.63213	.27859	.29723
10									1.60699	1.74491	.67590	.72059	.31608	.33510
11									1.89241	2.05095	.76624	.81289	.35429	.37366
12									2.22231	2.40872	.86060	.90942	.39320	.41293
13									2.61312	2.83952	.95935	1.01050	.43285	.45295
14									3.09257	3.38019	1.06292	1.11667	.47326	.49373
15									3.71302	4.10797	1.17181	1.22843	.51442	.53532
16									4.59366	5.22465	1.28660	1.34641	.55442	.57773
17									6.12683		1.40795	1.47134	.59926	.62101
18											1.53667	1.60408	.64299	.66520
19											1.67371	1.74570	.68764	.71032
20											1.82021	1.89745	.73324	.75642
21											1.97759	2.06089	.77985	.80355
22											2.14759	2.23795	.82751	.85175
23											2.33241	2.43123	.87626	.90107
24											2.53488	2.64385	.92616	.95156
25											2.75873	2.88018	.97726	1.00328
26											3.00902	3.14619	1.02963	1.05630
27											3.29284	3.45039	1.08331	1.11068
28											3.62059	3.80565	1.13839	1.16648
29											4.00840	4.23260	1.19495	1.22380
30											4.48333	4.76771	1.25304	1.28270
31											5.09621	5.48507	1.31276	1.34329
32											5.96158	6.57711	1.37425	1.40566
33											7.44730		1.43755	1.46992
34													1.50279	1.53619
35													1.57011	1.60459
36													1.63964	1.67527
37													1.71152	1.74839
38													1.78592	1.82412
39													1.86303	1.90266
40													1.94304	1.98421
41													2.02620	2.06903
42													2.11274	2.15738
43													2.20297	2.24957
44													2.29721	2.34594
45													2.39583	2.44691
46													2.49925	2.55292
47													2.60798	2.66451
48													2.72259	2.78230
49													2.84374	2.90701
50													2.97223	3.03952
51													3.10901	3.18086
52													3.25522	3.33229
53													3.41226	3.49537
54													3.58187	3.67204
55													3.76621	3.86477
56													3.96812	4.07677
57													4.19129	4.31234
58													4.44072	4.57738
59													4.72346	4.88032
60													5.04972	5.23384
61													5.43547	5.65830
62													5.90732	6.18950
63													6.51906	6.89961
64													7.37014	7.97539
65													8.82549	

Table A.8 (cont'd) Compandors for Signal with Laplacian Density (Mean = 0, Variance = 1)

BIT	8	9
LEVEL	256	512
M.S.E.	0.00007	0.00002
S.N.R.(DB)	41.64767	47.65760
ENTROPY	7.35847	8.38272

I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)	I	X(I)	Y(I)
1	0.000000	.0082360	65	1.4535	1.4695	1	0.0000000	.0041304	65	.60739	.61290	129	1.4618	1.4700	193	2.9151	2.9314
2	.016504	.024805	66	1.4863	1.5030	2	.0082683	.012415	66	.61841	.62394	130	1.4783	1.4866	194	2.9479	2.9646
3	.033138	.041504	67	1.5197	1.5366	3	.016570	.020733	67	.62949	.63505	131	1.4949	1.5033	195	2.9813	2.9982
4	.049903	.058335	68	1.5537	1.5708	4	.024904	.029083	68	.64062	.64621	132	1.5117	1.5201	196	3.0152	3.0326
5	.066801	.075301	69	1.5881	1.6056	5	.033270	.037466	69	.65181	.65743	133	1.5286	1.5371	197	3.0497	3.0671
6	.083836	.092404	70	1.6232	1.6409	6	.041670	.045883	70	.66306	.66871	134	1.5456	1.5542	198	3.0847	3.1025
7	.10101	.10965	71	1.6588	1.6769	7	.050103	.054333	71	.67437	.68005	135	1.5628	1.5714	199	3.1204	3.1385
8	.11832	.12703	72	1.6951	1.7134	8	.058570	.062816	72	.68574	.69145	136	1.5801	1.5888	200	3.1566	3.1750
9	.13577	.14456	73	1.7319	1.7506	9	.067071	.071334	73	.69718	.70292	137	1.5975	1.6063	201	3.1935	3.2121
10	.15337	.16223	74	1.7694	1.7885	10	.075606	.079885	74	.70867	.71444	138	1.6151	1.6240	202	3.2310	3.2500
11	.17112	.18005	75	1.8076	1.8270	11	.084175	.088473	75	.72023	.72603	139	1.6329	1.6418	203	3.2692	3.2885
12	.18902	.19802	76	1.8465	1.8663	12	.092780	.097095	76	.73185	.73769	140	1.6508	1.6598	204	3.3081	3.3278
13	.20707	.21615	77	1.8862	1.9063	13	.10142	.10575	77	.74354	.74940	141	1.6688	1.6799	205	3.3477	3.3678
14	.22527	.23443	78	1.9266	1.9470	14	.11009	.11444	78	.75529	.76118	142	1.6870	1.6962	206	3.3881	3.4085
15	.24363	.25287	79	1.9677	1.9886	15	.11880	.12317	79	.76710	.77303	143	1.7054	1.7146	207	3.4292	3.4501
16	.26215	.27147	80	2.0097	2.0310	16	.12755	.13194	80	.77898	.78495	144	1.7239	1.7332	208	3.4717	3.4925
17	.28084	.29024	81	2.0525	2.0743	17	.13633	.14074	81	.79093	.79693	145	1.7426	1.7520	209	3.5140	3.5357
18	.29969	.30918	82	2.0962	2.1186	18	.14515	.14957	82	.80295	.80898	146	1.7614	1.7709	210	3.5577	3.5799
19	.31871	.32828	83	2.1409	2.1635	19	.15401	.15845	83	.81503	.82110	147	1.7805	1.7900	211	3.6023	3.6250
20	.33790	.34756	84	2.1865	2.2096	20	.16290	.16736	84	.82718	.83329	148	1.7997	1.8093	212	3.6479	3.6711
21	.35727	.36702	85	2.2331	2.2567	21	.17183	.17631	85	.83941	.84554	149	1.8190	1.8288	213	3.6945	3.7187
22	.37681	.38665	86	2.2807	2.3049	22	.18080	.18530	86	.85170	.85787	150	1.8386	1.8484	214	3.7421	3.7663
23	.39654	.40647	87	2.3294	2.3542	23	.18980	.19432	87	.86407	.87028	151	1.8583	1.8682	215	3.7908	3.8156
24	.41645	.42648	88	2.3793	2.4047	24	.19885	.20338	88	.87651	.88275	152	1.8782	1.8882	216	3.8407	3.8661
25	.43655	.44668	89	2.4304	2.4564	25	.20793	.21249	89	.88902	.89530	153	1.8983	1.9085	217	3.8917	3.9177
26	.45685	.46707	90	2.4827	2.5095	26	.21705	.22163	90	.90160	.90792	154	1.9186	1.9288	218	3.9441	3.9707
27	.47734	.48766	91	2.5364	2.5637	27	.22622	.23081	91	.91426	.92062	155	1.9391	1.9494	219	3.9977	4.0251
28	.49803	.50845	92	2.5914	2.6195	28	.23542	.24003	92	.92700	.93340	156	1.9598	1.9702	220	4.0525	4.0808
29	.51892	.52944	93	2.6480	2.6769	29	.24466	.24929	93	.93982	.94625	157	1.9806	1.9912	221	4.1093	4.1381
30	.54002	.55065	94	2.7060	2.7357	30	.25394	.25860	94	.95271	.95918	158	2.0017	2.0124	222	4.1673	4.1969
31	.56133	.57207	95	2.7657	2.7962	31	.26326	.26794	95	.96569	.97219	159	2.0231	2.0338	223	4.2270	4.2575
32	.58286	.59371	96	2.8272	2.8586	32	.27263	.27732	96	.97875	.98529	160	2.0446	2.0554	224	4.2884	4.3198
33	.60461	.61557	97	2.8906	2.9228	33	.28203	.28675	97	.99186	.99846	161	2.0663	2.0773	225	4.3517	4.3840
34	.62659	.63766	98	2.9556	2.9890	34	.29148	.29622	98	1.0051	1.0117	162	2.0883	2.0994	226	4.4168	4.4502
35	.64879	.65998	99	3.0229	3.0574	35	.30097	.30573	99	1.0184	1.0251	163	2.1105	2.1217	227	4.4841	4.5185
36	.67123	.68255	100	3.0924	3.1280	36	.31050	.31528	100	1.0318	1.0385	164	2.1329	2.1443	228	4.5535	4.5891
37	.69391	.70534	101	3.1642	3.2011	37	.32008	.32488	101	1.0452	1.0520	165	2.1556	2.1671	229	4.62[illegible]	4.6622
38	.71684	.72839	102	3.2386	3.2768	38	.32969	.33452	102	1.0588	1.0656	166	2.1785	2.1901	230	4.6997	4.7378
39	.74001	.75170	103	3.3156	3.3557	39	.33936	.34420	103	1.0724	1.0793	167	2.2017	2.2134	231	4.7767	4.8163
40	.76345	.77528	104	3.3956	3.4367	40	[illegible]	.35393	104	1.0861	1.0930	168	2.2251	2.2370	232	4.8566	4.8977
41	.78714	.79909	105	3.4787	3.5215	41	.35881	.36371	105	1.0999	1.1069	169	2.2488	2.2608	233	4.9397	4.9825
42	.81110	.82318	106	3.5652	3.6098	42	.36861	.37353	106	1.1139	1.1208	170	2.2728	2.2849	234	5.0261	5.0707
43	.83533	.84756	107	3.6553	3.7019	43	.37845	.38339	107	1.1279	1.1349	171	2.2970	2.3092	235	5.1162	5.1628
44	.85985	.87221	108	3.7495	3.7982	44	.38834	.39330	108	1.1419	1.1490	172	2.3215	2.3339	236	5.2103	5.2590
45	.88465	.89716	109	3.8480	3.8990	45	.39827	.40326	109	1.1561	1.1633	173	2.3463	2.3588	237	5.3088	5.3595
46	.90975	.92240	110	3.9513	4.0050	46	.40825	.41326	110	1.1704	1.1776	174	2.3714	2.3841	238	5.4121	5.4657
47	.93514	.94795	111	4.0600	4.1144	47	.41828	.42331	111	1.1848	1.1920	175	2.3968	2.4096	239	5.5207	5.5771
48	.96084	.97381	112	4.1745	4.2341	48	.42835	.43341	112	1.1993	1.2065	176	2.4225	2.4355	240	5.6351	5.6947
49	.98686	.99999	113	4.2955	4.3587	49	.43848	.44356	113	1.2138	1.2212	177	2.4485	2.4617	241	5.7560	5.8192
50	1.0132	1.0265	114	4.4238	4.4910	50	.44865	.45376	114	1.2285	1.2359	178	2.4749	2.4882	242	5.8863	5.9514
51	1.0399	1.0533	115	4.5604	4.6322	51	.45887	.46400	115	1.2433	1.2507	179	2.5015	2.5150	243	6.0208	6.0925
52	1.0669	1.0805	116	4.7065	4.7838	52	.46914	.47430	116	1.2582	1.2657	180	2.5285	2.5422	244	6.1667	6.2438
53	1.0942	1.1081	117	4.8633	4.9463	53	.47947	.48464	117	1.2732	1.2807	181	2.5559	2.5697	245	6.3234	6.4063
54	1.1220	1.1360	118	5.0328	5.1227	54	.48984	.49504	118	1.2883	1.2958	182	2.5836	2.5976	246	6.4926	6.5825
55	1.1501	1.1642	119	5.2167	5.3151	55	.50026	.50549	119	1.3035	1.3111	183	2.6117	2.6259	247	6.6766	6.7767
56	1.1785	1.1929	120	5.4182	5.5262	56	.51073	.51599	120	1.3188	1.3265	184	2.6402	2.6545	248	6.8779	6.9861
57	1.2074	1.2219	121	5.6409	5.7617	57	.52126	.52654	121	1.3342	1.3419	185	2.6690	2.6836	249	7.1002	7.2208
58	1.2366	1.2514	122	5.8898	6.0762	58	.53183	.53714	122	1.3497	1.3575	186	2.6982	2.7130	250	7.3487	7.4849
59	1.2663	1.2812	123	6.1718	6.3283	59	.54247	.54780	123	1.3654	1.3732	187	2.7279	2.7429	251	7.6303	7.7865
60	1.2963	1.3115	124	6.4922	6.6907	60	.55315	.55851	124	1.3811	1.3891	188	2.7580	2.7732	252	7.9551	8.1382
61	1.3268	1.3423	125	6.8816	7.1036	61	.56389	.56928	125	1.3970	1.4050	189	2.7885	2.8039	253	8.3337	8.5601
62	1.3578	1.3735	126	7.3515	7.6123	62	.57468	.58010	126	1.4130	1.4211	190	2.8196	2.8351	254	8.8076	9.0873
63	1.3892	1.4051	127	7.9599	8.3341	63	.58553	.59097	127	1.4292	1.4373	191	2.8509	2.8667	255	9.4098	9.7904
64	1.4211	1.4372	128	8.8065	9.4011	64	.59643	.60191	128	1.4454	1.4536	192	2.8827	2.8988	256	10.256	10.849
			129	10.241											257	11.679	

Table A.9I Analytic Models for Some Often Used Quantizers for Unity Variance Probability Density Functions

Quantizer	Probability Density	Mean Square Error Model, f(n) n = number of bits 2^n = number of levels
Shannon	Arbitrary, Unitary Variance	$f(n) = 2^{-2n}\ ,\ n \geq 0$
Optimum Nonuniform	Gaussian N (0,1) $\frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$	$f(n) = \begin{cases} 2^{-1.5047n} & ,\ 0 \leq 2^n < 5 \\ 1.5253 \cdot 2^{-1.8274n} & ,\ 5 \leq 2^n < 36 \\ 2.2573 \cdot 2^{-1.9626n} & ,\ 36 \leq 2^n \leq 512 \end{cases}$
Optimum Nonuniform	Laplacian Unity Variance $\frac{1}{\sqrt{2}} \exp(-\sqrt{2}\lvert x\rvert)$	$f(n) = \begin{cases} 2^{-1.1711n} & ,\ 0 \leq 2^n < 5 \\ 2.0851 \cdot 2^{-1.7645n} & ,\ 5 \leq 2^n < 36 \\ 3.6308 \cdot 2^{-1.9572n} & ,\ 36 \leq 2^n \leq 512 \end{cases}$
Optimum Nonuniform	Gaussian N (0,1) $\frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$	$f(n) = \begin{cases} 2^{-1.5012n} & ,\ 0 \leq 2^n < 5 \\ 1.2477 \cdot 2^{-1.6883n} & ,\ 5 \leq 2^n < 36 \\ 1.5414 \cdot 2^{-1.7562n} & ,\ 36 \leq 2^n \leq 512 \end{cases}$
Optimum Uniform	Laplacian Unity Variance $\frac{1}{\sqrt{2}} \exp(-\sqrt{2}\lvert x\rvert)$	$f(n) = \begin{cases} 2^{-1.1619n} & ,\ 0 \leq 2^n < 5 \\ 1.4156 \cdot 2^{-1.4518n} & ,\ 5 \leq 2^n < 36 \\ 2.1969 \cdot 2^{-1.5944n} & ,\ 36 \leq 2^n \leq 512 \end{cases}$
Approximate Optimum Nonuniform (Compandor)	Gaussian N (0,1) $\frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$	$f(n) = \begin{cases} 2^{-1.4864n} & ,\ 0 \leq 2^n < 5 \\ 1.5597 \cdot 2^{-1.8239n} & ,\ 5 \leq 2^n < 36 \\ 0.2677 \cdot 2^{-1.357\text{[illegible]}n} & ,\ 36 \leq 2^n \leq 512 \end{cases}$
Approximate Optimum Nonuniform (Compandor)	Laplacian Unity Variance $\frac{1}{\sqrt{2}} \exp(-\sqrt{2}\lvert x\rvert)$	$f(n) = \begin{cases} 2^{-1.1491n} & ,\ 0 \leq 2^n < 5 \\ 2.2164 \cdot 2^{-1.7710n} & ,\ 5 \leq 2^n < 36 \\ 4.2461 \cdot 2^{-1.9928n} & ,\ 36 \leq 2^n \leq 512 \end{cases}$

Table A.9II Formulas for h(x) for Different Quantizers

Quantizer	$h(x) = f'(x)^{-1}$
Shannon	$h(x) = -\frac{1}{2}\log_2(-x/2\log_e 2)$ all $x < 0$
Optimum Nonuniform (Gaussian Density)	$h(x) = \begin{cases} 0.0403 - 0.6646\log_2(-x), & -1.0430 \le x < -0.1020 \\ 0.5199 - 0.5472\log_2(-x), & -0.1020 \le x < -0.0027 \\ 0.8247 - 0.5195\log_2(-x), & -0.0027 \le x \le -1.48 \times 10^{-5} \end{cases}$
Optimum Nonuniform (Laplacian Density)	$h(x) = \begin{cases} -0.2569 - 0.8539\log_2(-x), & -0.8117 \le x < -0.1490 \\ 0.7654 - 0.5667\log_2(-x), & -0.1490 \le x < -0.0044 \\ 1.1753 - 0.5109\log_2(-x), & -0.0044 \le x \le -2.45 \times 10^{-5} \end{cases}$
Optimum Uniform (Gaussian Density)	$h(x) = \begin{cases} 0.0382 - 0.6661\log_2(-x), & -1.0406 \le x < -0.0965 \\ 0.3234 - 0.5923\log_2(-x), & -0.0965 < x < -0.0035 \\ 0.5170 - 0.5694\log_2(-x), & -0.0035 \le x \le -3.28 \times 10^{-5} \end{cases}$
Optimum Uniform (Laplacian Density)	$h(x) = \begin{cases} -0.2688 - 0.8607\log_2(-x), & -0.8054 \le x < -0.1377 \\ 0.3516 - 0.6888\log_2(-x), & -0.1377 \le x < -0.0080 \\ 0.8026 - 0.6272\log_2(-x), & -0.0080 \le x \le -1.16 \times 10^{-4} \end{cases}$
Approximate Optimum Nonuniform (Gaussian Density)	$h(x) = \begin{cases} 0.0290 - 0.6728\log_2(-x), & -1.0303 \le x < -0.1047 \\ 0.5371 - 0.5483\log_2(-x), & -0.1047 \le x < -0.0019 \\ -1.4658 - 0.7368\log_2(-x), & -0.0019 \le x \le -5.29 \times 10^{-5} \end{cases}$
Approximate Optimum Nonuniform (Laplacian Density)	$h(x) = \begin{cases} -0.2857 - 0.8702\log_2(-x), & -0.7965 \le x < -0.1573 \\ 0.8154 - 0.5647\log_2(-x), & -0.1573 \le x < -0.0046 \\ 4.2461 - 0.5018\log_2(-x), & -0.0046 \le x \le -2.34 \times 10^{-5} \end{cases}$

Table A.10 Optimum Nonuniform Quantizers with Quantization Levels N = 2,...,36 for Signal with Gaussian Density (Mean = 0, Variance = 1)

LEVEL	2		3		4		5		6		7		8	
M.S.E.	.36338		.19017		.11748		.079941		.057978		.044000		.034548	
S.N.R.(DB)	4.3964		7.2085		9.3003		10.972		12.367		13.565		14.616	
ENTROPY	1.0000		1.5358		1.911[illegible]		2.2029		2.4428		2.6469		2.8248	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.79788	.61201	0.00000	0.00000	.45279	.38230	0.00000	0.00000	.31773	.28030	0.00000	0.00000	.24511
2				1.2240	.98162	1.5104	1.2444	.76459	.65894	1.0001	.87440	.56060	.50059	.75605
3								1.7242	1.4469	1.8936	1.6108	1.1882	1.0500	1.3440
4												2.0334	1.7480	2.1520

LEVEL	9		10		11		12		13		14		15	
M.S.E.	.027853		.022937		.019220		.016340		.014063		.012232		.010737	
S.N.R.(DB)	15.551		16.395		17.163		17.868		18.519		19.125		19.691	
ENTROPY	2.9826		3.1245		3.2536		3.3716		3.4806		3.5819		3.6765	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	.22184	0.00000	0.00000	.19965	.18376	0.00000	0.00000	.16847	.15692	0.00000	0.00000	.14574	.13696	0.00000
2	.68128	.44368	.40479	.60992	.55999	.36751	.34020	.51193	.47620	.31383	.29358	.44142	.41442	.27393
3	1.1977	.91886	.83393	1.0579	.96571	.75246	.69442	.87690	.81274	.63836	.59601	.75059	.70312	.55490
4	1.8656	1.4765	1.3247	1.5914	1.4359	1.1789	1.0814	1.2859	1.1843	.98710	.91821	1.0858	1.0132	.85132
5		2.2547	1.9683	2.3452	2.0593	1.6928	1.5345	1.7832	1.6231	1.3314	1.2768	1.4677	1.3607	1.1751
6						2.4259	2.1409	2.4986	2.2147	1.8647	1.7033	1.9388	1.7765	1.5463
7										2.5647	2.2820	2.6253	2.3439	2.0067
8														2.6811

LEVEL	16		17		18		19		20		21		22	
M.S.E.	.0095010		.0084669		.0075932		.0068482		.0062078		.0056533		.0051700	
S.N.R.(DB)	20.222		20.723		21.196		21.644		22.071		22.477		22.865	
ENTROPY	3.7652		3.8466		3.9275		4.0023		4.0773		4.1410		4.2056	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.12843	.12154	0.00000	0.00000	.11481	.10925	0.00000	0.00000	.10381	.099228	0.00000	0.00000	.94738
2	.25830	.38616	.36706	.24307	.23064	.34647	.32952	.21850	.20837	.31293	.29900	.19845	.19005	.28535
3	.52255	.65693	.62028	.49104	.46549	.58450	.55529	.44053	.41983	.52671	.50286	.39955	.38242	.47948
4	.79975	.94256	.88769	.74950	.70932	.83413	.79102	.67004	.63778	.74883	.71399	.60617	.57964	.67982
5	1.0995	1.2565	1.1785	1.0253	.96828	1.1024	1.0426	.91193	.86643	.98600	.93638	.82179	.78476	.88968
6	1.4374	1.6183	1.5080	1.3312	1.2513	1.4062	1.3187	1.1732	1.1114	1.2389	1.1756	1.0510	1.0016	1.1135
7	1.8433	2.0593	1.9060	1.6848	1.5733	1.7464	1.6340	1.4642	1.3814	1.5238	1.4399	1.3002	1.2355	1.3576
8	2.4011	2.7328	2.4542	2.1273	1.9638	2.1813	2.0177	1.8937	1.6906	1.8574	1.7437	1.5797	1.4949	1.6321
9				2.7810	2.5037	2.8261	2.5501	2.2317	2.0643	2.2791	2.1158	1.9078	1.7937	1.9553
10								2.8684	2.5937	2.9083	2.6349	2.3237	2.1606	2.3659
11												2.9460	2.6738	2.9817

Table A.10 (cont'd) Optimum Nonuniform Quantizers with Quantization Levels N = 2,...,36 for Signal with Gaussian Density (Mean = 0, Variance = 1)

LEVEL	23		24		25		26		27		28		29	
M.S.E.	.0047462		.0043725		.0040413		.0037464		.0034826		.0032458		.0030323	
S.N.R.(DB)	23.237		23.593		23.935		24.264		24.581		24.887		25.182	
ENTROPY	4.2675		4.3267		4.3837		4.4384		4.4911		4.5420		4.5911	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	.090899	0.00000	0.00000	.087128	.083864	0.00000	0.00000	.080654	.077844	0.00000	0.00000	.075078	.072635	0.00000
2	.27371	.18180	.17470	.26227	.25348	.16773	.16166	.24266	.23416	.15569	.15044	.22580	.21842	.14527
3	.45963	.36561	.35120	.44013	.42333	.33704	.32474	.40681	.39242	.31264	.30202	.37824	.36576	.29156
4	.65103	.55364	.53144	.62274	.59852	.50962	.49076	.57469	.55402	.47219	.45595	.53366	.51580	.43995
5	.85077	.74841	.71764	.81254	.78007	.68740	.66142	.74813	.72060	.63584	.61358	.69348	.66981	.59163
6	1.0626	.95312	.91258	1.0126	.97067	.87273	.83877	.92940	.89414	.80535	.77646	.85942	.82931	.74797
7	1.2918	1.1721	1.1199	1.2272	1.1739	1.0686	1.0254	1.1214	1.0772	.98292	.94652	1.0336	.99619	.91063
8	1.5466	1.4116	1.3448	1.4624	1.3949	1.2792	1.2249	1.3283	1.2732	1.1715	1.1263	1.2189	1.1729	1.0817
9	1.8408	1.6816	1.5954	1.7283	1.6416	1.5105	1.4423	1.5561	1.4872	1.3750	1.3191	1.4193	1.3628	1.2640
10	2.2029	2.0001	1.8854	2.0426	1.9277	1.7725	1.6854	1.8146	1.7270	1.5995	1.5300	1.6407	1.5708	1.4615
11	2.7107	2.4058	2.2431	2.4437	2.2813	2.0829	1.9679	2.1213	2.0062	1.8546	1.7668	1.8928	1.8047	1.6501
12		3.0156	2.7458	3.0479	2.7792	2.4797	2.3177	2.5141	2.3524	2.1578	2.0428	2.1928	2.0778	1.9293
13						3.0787	2.8111	3.1081	2.8416	2.5470	2.3855	2.5744	2.4174	2.2263
14										3.1363	2.8709	3.1634	2.8959	2.6085
15														3.1893

LEVEL	30		31		32		33		34		35		36	
M.S.E.	.0028393		.0026641		.0025047		.0023591		.0022259		.0021037		.0019913	
S.N.R.(DB)	25.468		25.744		26.012		26.272		26.525		26.770		27.009	
ENTROPY	4.6386		4.6846		4.7291		4.7723		4.8142		4.8550		4.8946	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.070226	.068081	0.00000	0.00000	.065965	.064067	0.00000	0.00000	.062193	.060503	0.00000	0.00000	.058832
2	.14068	.21114	.20467	.13616	.13212	.19828	.19255	.12813	.12455	.18690	.18180	.12100	.11780	.17676
3	.28230	.35346	.34253	.27316	.26502	.33175	.32210	.25697	.24974	.31257	.30400	.24260	.23614	.29552
4	.42583	.49819	.48260	.41189	.39948	.46721	.45347	.38723	.37624	.43990	.42771	.36539	.35559	.41566
5	.57233	.64647	.62589	.55329	.53639	.60557	.58740	.51971	.50478	.56964	.55364	.49003	.47675	.53783
6	.72305	.79962	.77359	.69847	.67674	.74789	.72513	.65528	.63615	.70264	.68256	.61725	.60027	.66271
7	.87936	.95978	.92714	.84870	.82166	.89542	.86749	.79496	.77127	.83989	.81536	.74786	.72691	.79111
8	1.0434	1.1275	1.0884	1.0056	.97258	1.0497	1.0160	.93999	.91125	.98259	.95312	.88284	.85735	.92398
9	1.2173	1.3071	1.2596	1.1711	1.1313	1.2128	1.1723	1.0919	1.0574	1.1323	1.0972	1.0234	.99325	1.0625
10	1.4045	1.5019	1.4443	1.3481	1.3001	1.3874	1.3389	1.2527	1.2116	1.2909	1.2493	1.1710	1.1353	1.2081
11	1.6098	1.7177	1.6471	1.5404	1.4824	1.5773	1.5189	1.4250	1.3760	1.4611	1.4117	1.3276	1.2855	1.3628
12	1.8410	1.9642	1.8757	1.7538	1.6828	1.7883	1.7171	1.6127	1.5539	1.6467	1.5876	1.4958	1.4460	1.5292
13	2.1113	2.2584	2.1434	1.9977	1.9091	2.0298	1.9411	1.8215	1.7501	1.8534	1.7818	1.6794	1.6201	1.7110
14	2.4479	2.6375	2.4772	2.2892	2.1743	2.3188	2.2040	2.0607	1.9720	2.0905	2.0017	1.8842	1.8124	1.9138
15	2.9259	3.2143	2.9518	2.6653	2.5054	2.6921	2.5326	2.3473	2.2326	2.3748	2.2602	2.1192	2.0303	2.1469
16				3.2384	2.9768	3.2616	3.0010	2.7179	2.5588	2.7428	2.5841	2.4013	2.2869	2.4269
17								3.2840	3.0242	3.3056	3.0468	2.7669	2.6086	2.7902
18												3.3264	3.0685	3.3469

Table A.11 Optimum Nonuniform Quantizers with Quantization Levels N = 2,...,36 for Signal with Laplacian Density (Mean = 0, Variance = 1)

LEVEL	2		3		4		5		6		7		8	
M.S.E.	.50000		.26424		.17619		.11981		.059868		.068088		.054476	
S.N.R.(DB)	3.0103		5.7800		7.5401		9.2152		10.464		11.669		12.638	
ENTROPY	1.1000		1.3169		1.7282		1.9466		2.2071		2.3745		2.5654	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.70711	.70712	0.00000	0.00000	.41976	.41977	0.00000	0.00000	.29979	.29979	0.00000	0.00000	.23341
2				.14142	1.1269	1.8340	1.5467	.83953	.71956	1.1393	1.0194	.59958	.53321	.83299
3								2.2538	1.8464	2.5535	2.1462	1.4391	1.2528	1.6725
4												2.8533	2.3797	3.0868

LEVEL	9		10		11		12		13		14		15	
M.S.E.	.043850		.036545		.030580		.026213		.022536		.019720		.017293	
S.N.R.(DB)	13.580		14.372		15.146		15.815		16.471		17.051		17.621	
ENTROPY	2.7011		2.8519		2.9661		3.0907		3.1893		3.2955		3.3822	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	.23342	0.00000	0.00000	.19118	.19118	0.00000	0.00000	.16192	.16192	0.00000	0.00000	.14044	.14045	0.00000
2	.76663	.46683	.42460	.65801	.61578	.38236	.35310	.54429	.51503	.32385	.30237	.46429	.44282	.28989
3	1.4862	1.0664	.95761	1.2576	1.1490	.84920	.77771	1.0111	.93964	.70622	.65548	.84667	.79594	.60474
4	2.6131	1.9060	1.6774	2.0971	1.8686	1.4488	1.3109	1.6107	1.4729	1.1731	1.0801	1.3135	1.2206	.98712
5		3.3202	2.8043	3.5114	2.9955	2.2883	2.0305	2.4503	2.1924	1.7727	1.6133	1.9131	1.7538	1.4540
6						3.7026	3.1574	3.8645	3.3193	2.6122	2.3329	2.7527	2.4733	2.0536
7										4.0264	3.4598	4.1669	3.6002	2.8931
8														4.3073

LEVEL	16		17		18		19		20		21		22	
M.S.E.	.015373		.013688		.012320		.011104		.010094		.0091873		.0084219	
S.N.R.(DB)	18.133		18.636		19.094		19.545		19.959		20.368		20.746	
ENTROPY	3.4747		3.5521		3.6341		3.7040		3.7776		3.8413		3.9081	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.12400	.12401	0.00000	0.00000	.11102	.11102	0.00000	0.00000	.10049	.10050	0.00000	0.00000	.091795
2	.26445	.40490	.38347	.24801	.23503	.35903	.34605	.22204	.21151	.32253	.31202	.20099	.19229	.29279
3	.56683	.72876	.69085	.52891	.49949	.63934	.61052	.47006	.44655	.57056	.54706	.42304	.40382	.51484
4	.91995	1.1111	1.0440	.85278	.80188	.96380	.91291	.75097	.73102	.85148	.81153	.67107	.63686	.76288
5	1.3446	1.5780	1.4686	1.2352	1.1550	1.3462	1.2660	1.0748	1.0134	1.1753	1.1139	.95199	.90334	1.0438
6	1.8778	2.1776	2.0018	1.7020	1.5796	1.8130	1.6907	1.4572	1.3665	1.5577	1.4671	1.2759	1.2057	1.3677
7	2.5974	3.0171	2.7214	2.3016	2.1129	2.4126	2.2239	1.9241	1.7912	2.0246	1.8917	1.6583	1.5589	1.7501
8	3.7243	4.4314	3.8483	3.1412	2.8324	3.2522	2.9435	2.5237	2.3244	2.6242	2.4249	2.1251	1.9835	2.2149
9				4.5554	3.9573	4.6664	4.0704	3.3632	3.0440	3.4637	3.1445	2.7247	2.5167	2.8165
10								4.7775	4.1709	4.8780	4.2714	3.5643	3.2363	3.6561
11												4.9785	4.3632	5.0703

Table A.11 (cont'd) Optimum Nonuniform Quantizers with Quantization Levels N = 2,...,36
for Signal with Laplacian Density (Mean = 0, Variance = 1)

LEVEL	23	24	25	26	27	28	29
M.S.E	.0077275	.0071332	.0065898	.0061192	.0056861	.0053071	.0049563
S.N.R.(DB)	21.120	21.467	21.811	22.133	22.452	22.751	23.048
ENTROPY	3.9666	4.0277	4.0819	4.1382	4.1886	4.2408	4.2879

I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	.091798	0.00000	0.00000	.084484	.084487	0.00000	0.00000	.078253	.078257	0.00000	0.00000	.072880	.072883	0.00000
2	.28410	.18360	.17629	.26808	.26078	.16897	.16274	.24723	.24101	.15651	.15114	.22940	.22403	.14577
3	.49563	.38460	.36859	.46909	.45309	.35258	.33904	.43064	.41731	.32550	.31389	.39839	.38679	.30229
4	.73068	.60665	.58012	.69115	.66462	.55359	.53135	.63186	.60963	.50911	.49020	.58201	.56310	.47128
5	.99516	.85469	.81518	.93919	.89968	.77565	.74289	.85392	.82117	.71013	.68252	.78303	.75542	.65491
6	1.2976	1.1356	1.0797	1.2201	1.1642	1.0237	.97795	1.1020	1.0562	.93220	.89407	1.0051	.96698	.85593
7	1.6507	1.4595	1.3821	1.5440	1.4666	1.3046	1.2424	1.3829	1.3207	1.1803	1.1291	1.2532	1.2020	1.0780
8	2.0753	1.8419	1.7352	1.9264	1.8197	1.6285	1.5448	1.7068	1.6231	1.4612	1.3936	1.5341	1.4665	1.3261
9	2.6085	2.3087	2.1598	2.3932	2.2443	2.0109	1.8980	2.0892	1.9763	1.7851	1.6960	1.8580	1.7689	1.6070
10	3.3281	2.9083	2.6931	2.9929	2.7776	2.4778	2.3226	2.5560	2.4009	2.1675	2.0492	2.2404	2.1221	1.9309
11	4.4550	3.7479	3.4126	3.8324	3.4971	3.0774	2.8558	3.1556	2.9341	2.6343	2.4738	2.7072	2.5467	2.3133
12		5.1621	4.5395	5.2466	4.6240	3.9169	3.5754	3.9952	3.6537	3.2339	3.0070	3.3068	3.0799	2.7801
13						5.3311	4.7023	5.4094	4.7806	4.0735	3.7266	4.1464	3.7995	3.3797
14										5.4877	4.8535	5.5606	4.9264	4.2193
15														5.6335

LEVEL	30	31	32	33	34	35	36
M.S.E.	.0046466	.0043585	.0041022	.0038627	.0036481	.0034469	.0032655
S.N.R.(DB)	23.329	23.607	23.870	24.131	24.379	24.626	24.860
ENTROPY	4.3366	4.3808	4.4264	4.4680	4.5109	4.5503	4.5907

I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.058198	.068201	0.00000	0.00000	.064082	.064085	0.00000	0.00000	.060435	.060439	0.00000	0.00000	.057182
2	.14108	.21397	.20929	.13640	.13229	.20049	.19638	.12817	.12452	.18861	.18497	.12088	.11762	.17806
3	.29223	.37050	.36045	.28218	.27338	.34627	.33748	.26458	.25682	.32503	.31727	.24906	.24216	.30625
4	.45500	.53950	.52322	.43871	.42454	.50281	.48864	.41037	.39793	.47082	.45838	.38548	.37447	.44268
5	.63131	.72312	.69953	.60772	.58731	.67181	.65142	.56691	.54310	.62737	.60955	.53128	.51558	.58848
6	.82364	.92415	.89107	.79135	.76363	.85545	.82774	.73592	.71168	.79638	.77234	.68783	.66676	.74503
7	1.0352	1.1462	1.1034	.99238	.95597	1.0565	1.0201	.91956	.88820	.98002	.94867	.85684	.82954	.91405
8	1.2703	1.3943	1.3385	1.2145	1.1675	1.2786	1.2316	1.1206	1.0805	1.1811	1.1410	1.0405	1.0059	1.0977
9	1.5348	1.6752	1.6030	1.4625	1.4026	1.5266	1.4667	1.3427	1.2921	1.4031	1.3526	1.2415	1.1982	1.2987
10	1.8372	1.9991	1.9054	1.7435	1.6671	1.8076	1.7312	1.5907	1.5272	1.6512	1.5877	1.4636	1.4098	1.5208
11	2.1903	2.3815	2.2585	2.0673	1.9695	2.1314	2.0336	1.8717	1.7917	1.9321	1.8522	1.7117	1.6549	1.7689
12	2.6149	2.8484	2.6832	2.4497	2.3227	2.5138	2.3868	2.1956	2.0941	2.2560	2.1546	1.9926	1.9094	2.0498
13	3.1482	3.4480	3.2164	2.9166	2.7473	2.9807	2.8114	2.5780	2.4472	2.6384	2.5077	2.3165	2.2118	2.3737
14	3.8677	4.2875	3.9360	3.5162	3.2805	3.5803	3.3446	3.0168	2.8719	3.1053	2.9323	2.6989	2.5649	2.7561
15	4.9946	5.7017	5.0629	4.3557	4.0001	4.4199	4.0642	3.6444	3.4051	3.7049	3.4656	3.1658	2.9896	3.2230
16				5.7700	5.1270	5.8341	5.1911	4.4840	4.1247	4.5544	4.1852	3.7654	3.5228	3.8226
17								5.8982	5.2516	5.9587	5.3120	4.6049	4.2424	4.6621
18												6.0191	5.3693	6.0764

Table A.12 Optimum Uniform Quantizers with Quantization Levels N = 2,...,36 for Signal with Gaussian Density (Mean = 0, Variance = 1)

NO. OF OUTPUT LEVELS	STEP SIZE	M.S.E.	S.N.R.(DB)	ENTROPY
2	1.5958	.36338	4.3964	1.0000
3	1.2240	.19017	7.2085	1.5358
4	.99569	.11885	9.2502	1.9037
5	.84299	.082174	10.857	2.1831
6	.73343	.060657	12.171	2.4085
7	.65077	.046860	13.292	2.5976
8	.58602	.037449	14.267	2.7606
9	.53382	.030696	15.129	2.9039
10	.49078	.025687	15.903	3.0319
11	.45462	.021856	16.604	3.1476
12	.42379	.018853	17.246	3.2532
13	.39716	.016453	17.838	3.3503
14	.37391	.014500	18.386	3.4402
15	.35341	.012889	18.898	3.5240
16	.33520	.011543	19.377	3.6024
17	.31890	.010405	19.828	3.6761
18	.30423	.0094339	20.253	3.7456
19	.29093	.0085979	20.656	3.8115
20	.27883	.0078727	21.039	3.8740
21	.26776	.0072390	21.403	3.9334
22	.25760	.0066818	21.751	3.9902
23	.24823	.0061890	22.084	4.0445
24	.23957	.0057510	22.403	4.0965
25	.23153	.0053596	22.709	4.1464
26	.22405	.0050084	23.003	4.1944
27	.21707	.0046920	23.286	4.2407
28	.21054	.0044058	23.560	4.2853
29	.20442	.0041460	23.824	4.3283
30	.19867	.0039094	24.079	4.3699
31	.19325	.0036932	24.326	4.4102
32	.18814	.0034952	24.565	4.4493
33	.18331	.0033133	24.797	4.4871
34	.17874	.0031458	25.023	4.5239
35	.17441	.0029912	25.242	4.5596
36	.17029	.0028481	25.454	4.5943

Table A.13 Optimum Uniform Quantizers with Quantization Levels N = 2,...,36 for Signal with Laplacian Density (Mean = 0, Variance = 1)

NO. OF OUTPUT LEVELS	STEP SIZE	M.S.E.	S.N.R.(DB)	ENTROPY
2	1.4142	.50000	3.0103	1.0000
3	1.4142	.26424	5.7800	1.3169
4	1.0874	.19630	7.0707	1.7507
5	1.0245	.13300	8.7616	1.8649
6	.87070	.10951	9.6056	2.1254
7	.82175	.083087	10.805	2.2073
8	.73093	.071748	11.442	2.3919
9	.69397	.057992	12.366	2.4608
10	.63346	.051488	12.883	2.6029
11	.60481	.043319	13.633	2.6637
12	.56131	.039174	14.070	2.7789
13	.53847	.033881	14.700	2.8336
14	.50555	.031044	15.080	2.9303
15	.48688	.027395	15.623	2.9802
16	.46100	.025351	15.960	3.0634
17	.44544	.022716	16.437	3.1093
18	.42449	.021185	16.740	3.1823
19	.41130	.019212	17.164	3.2248
20	.39396	.018029	17.440	3.2898
21	.38261	.016509	17.823	3.3294
22	.36800	.015574	18.076	3.3879
23	.35813	.014374	18.424	3.4249
24	.34562	.013619	18.659	3.4782
25	.33694	.012654	18.978	3.5130
26	.32611	.012033	19.196	3.5618
27	.31841	.011244	19.491	3.5947
28	.30892	.010727	19.695	3.6398
29	.30204	.010072	19.969	3.6708
30	.29365	.0096354	20.161	3.7127
31	.28746	.0090851	20.417	3.7422
32	.27999	.0087132	20.598	3.7813
33	.27439	.0082459	20.838	3.8094
34	.26768	.0079258	21.010	3.8460
35	.26258	.0075251	21.235	3.8728
36	.25652	.0072473	21.398	3.9073

Table A.14 Compandors with Quantization Levels N = 2,...,36 for Signal with Gaussian Density (Mean = 0, Variance = 1)

LEVEL	2		3		4		5		6		7		8	
M.S.E.	.36838		.19395		.12035		.082139		.059687		.045355		.03564	
S.N.R.(DB)	4.3371		7.1231		9.1954		10.855		12.241		13.434		14.481	
ENTROPY	1.0000		1.5675		1.9587		2.2578		2.5007		2.7056		2.8830	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.72720	.53576	0.00000	0.00000	.42426	.35172	0.00000	0.00000	.30071	.26282	0.00000	0.00000	.23355
2	1.6215		1.9056	1.1300	.87607	1.3983	1.1217	.71874	.61084	.94207	.81451	.53188	.47144	.71862
3					2.0971		2.2425	1.5976	1.3122	1.7554	1.4672	1.1208	.98145	1.2692
4									2.3601		2.4588	1.8855	1.5972	1.9959
5													2.5438	

LEVEL	9		10		11		12		13		14		15	
M.S.E.	.028747		.023678		.019842		.016868		.014516		.012623		.011077	
S.N.R.(DB)	15.414		16.256		17.024		17.729		18.382		18.988		19.556	
ENTROPY	3.0398		3.1803		3.3077		3.4242		3.5317		3.6314		3.7244	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	.21022	0.00000	0.00000	.19119	.17536	0.00000	0.00000	.16197	.15051	0.00000	0.00000	.14058	.13188	0.00000
2	.64363	.42360	.38474	.58324	.53349	.35253	.32538	.49174	.45618	.30216	.28208	.42551	.39877	.26454
3	1.1224	.87461	.78974	1.0082	.91629	.72048	.66277	.84054	.77688	.61389	.57191	.72253	.67553	.53543
4	1.7091	1.3956	1.2440	1.5055	1.3506	1.1246	1.0277	1.2276	1.1269	.94706	.87882	1.0426	.97082	.82019
5	2.6186	2.0917	1.8069	2.1762	1.8938	1.6025	1.4455	1.6891	1.5308	1.3199	1.2164	1.4033	1.2976	1.1292
6			2.6853		2.7455	2.2517	1.9718	2.3199	2.0425	1.7674	1.6982	1.8386	1.6790	1.4793
7							2.8003		2.8506	2.3820	2.1072	2.4390	2.1666	1.9040
8											2.8972		2.9404	2.4917

LEVEL	16		17		18		19		20		21		22	
M.S.E.	.0097991		.0087297		.0078261		.0070557		.0063935		.0058203		.0053207	
S.N.R.(DB)	20.088		20.590		21.065		21.515		21.943		22.351		22.740	
ENTROPY	3.8117		3.8938		3.9713		4.0448		4.1147		4.1812		4.2448	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.12421	.11739	0.00000	0.00000	.11129	.10579	0.00000	0.00000	.10082	.096295	0.00000	0.00000	.092162
2	.24907	.37524	.35437	.23533	.22304	.33573	.31898	.21199	.20199	.30384	.29009	.19289	.18459	.27754
3	.50343	.63445	.59821	.47511	.44986	.56598	.53713	.42721	.40676	.51114	.48759	.38821	.37130	.46616
4	.76920	.90879	.85458	.72441	.68471	.80674	.76419	.64927	.61741	.72606	.69167	.58861	.56244	.66049
5	1.0545	1.2082	1.1313	.98977	.93293	1.0643	1.0053	.88259	.83764	.95282	.90587	.79725	.76071	.86355
6	1.3720	1.5491	1.4405	1.2808	1.2020	1.3479	1.2676	1.1332	1.0724	1.1973	1.1349	1.0182	.96952	1.0793
7	1.7441	1.9642	1.8044	1.6136	1.5040	1.6734	1.5630	1.4102	1.3287	1.4684	1.3859	1.2571	1.1935	1.3132
8	2.2217	2.5406	2.2729	2.0202	1.8605	2.0723	1.9129	1.7291	1.6182	1.7813	1.6700	1.5228	1.4395	1.5740
9	2.9808		3.0187	2.5862	2.3207	2.6290	2.3656	2.1211	1.9620	2.1669	2.0082	1.8306	1.7188	1.8766
10					3.0544		3.0881	2.6692	2.4078	2.7071	2.4477	2.2102	2.0518	2.2510
11									3.1200		3.1503	2.7430	2.4854	2.7771
12													3.1792	

Table A.14 (cont'd) Compandors with Quantization Levels N = 2,...,36 for Signal with Gaussian Density (Mean = 0, Variance = 1)

LEVELS	23		24		25		26		27		28		29	
M.S.E.	.0048827		.0044966		.0041544		.0038499		.0035775		.0033331		.0031128	
S.N.R.(DB)	23.113		23.471		23.815		24.146		24.464		24.772		25.069	
ENTROPY	4.3056		4.3639		4.4199		4.4738		4.5257		4.5758		4.6241	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	.088371	0.00000	0.00000	.084881	.081658	0.00000	0.00000	.078672	.075898	0.00000	0.00000	.073314	.070901	0.00000
2	.26604	.17697	.16997	.25546	.24571	.16350	.15751	.23667	.22828	.15194	.14676	.22067	.21318	.14192
3	.44656	.35582	.34159	.42857	.41199	.32847	.31633	.39667	.38245	.30507	.29458	.36924	.35692	.28480
4	.63208	.53855	.51664	.60607	.58217	.49648	.47787	.56013	.53973	.46062	.44459	.52079	.50316	.42965
5	.82517	.72750	.69715	.79020	.75818	.66931	.64368	.72875	.70159	.61998	.59801	.67644	.65308	.57758
6	1.0291	.92555	.88559	.98375	.94241	.84910	.81562	.90459	.86983	.78478	.75628	.83776	.80857	.72985
7	1.2485	1.1365	1.0852	1.1905	1.1380	1.0385	.99602	1.0903	1.0467	.95703	.92115	1.0067	.96984	.88799
8	1.4901	1.3660	1.3003	1.4157	1.3493	1.2413	1.1880	1.2895	1.2353	1.1393	1.0948	1.1859	1.1405	1.0539
9	1.7648	1.6223	1.5378	1.6680	1.5830	1.4628	1.3957	1.5075	1.4398	1.3352	1.2803	1.3787	1.3231	1.2301
10	2.0931	1.9203	1.8084	1.9618	1.8497	1.7113	1.6259	1.7524	1.6667	1.5499	1.4818	1.5906	1.5218	1.4201
11	2.5213	2.2897	2.1323	2.3265	2.1696	2.0011	1.8891	2.0386	1.9266	1.7916	1.7056	1.8290	1.7428	1.6290
12	3.2068	2.8095	2.5554	2.8504	2.5879	2.3616	2.2051	2.3950	2.2391	2.0744	1.9624	2.1086	1.9967	1.8647
13			3.2232		3.2585	2.8699	2.6150	2.6981	2.6488	2.4270	2.2716	2.4576	2.3207	2.1413
14							3.2827		3.3060	2.9252	2.6773	2.9512	2.7047	2.4870
15											3.3285		3.3501	2.9762

LEVELS	30		31		32		33		34		35		36	
M.S.E.	.0029136		.0027329		.0025685		.0024185		.0022812		.0021553		.0020395	
S.N.R.(DB)	25.356		25.634		25.903		26.165		26.418		26.665		26.905	
ENTROPY	4.6709		4.7162		4.7601		4.8027		4.8441		4.8843		4.9234	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.068642	.066524	0.00000	0.00000	.064533	.062658	0.00000	0.00000	.060889	.059218	0.00000	0.00000	.057637
2	.13739	.20636	.19997	.13315	.12916	.19396	.18830	.12540	.12185	.18297	.17793	.11851	.11534	.17317
3	.27566	.34540	.33461	.26708	.25903	.32448	.31495	.25146	.24432	.30597	.29749	.23757	.23119	.28947
4	.41570	.48670	.47130	.40264	3.9039	.45687	.44330	.37887	.36801	.43053	.41848	.35777	.34909	.40710
5	.55853	.63132	.61101	.54072	.52403	.59199	.57414	.50836	.49362	.55736	.54156	.47972	.46659	.52664
.6	.70526	.78050	.75481	.68232	.66087	.73081	.70835	.64076	.62187	.68726	.66743	.60409	.58733	.64874
7	.85724	.93571	.90402	.82864	.80197	.87451	.84695	.77703	.75364	.82115	.79694	.73167	.71099	.77416
8	1.0161	1.0988	1.0603	.98115	.94861	1.0245	.99121	.91828	.88993	.96014	.93106	.86335	.83539	.90378
9	1.1841	1.2723	1.2257	1.1417	1.1024	1.1876	1.1427	1.0659	1.0320	1.1056	1.0710	1.0002	.97050	1.0387
10	1.3640	1.4597	1.4031	1.3127	1.2654	1.3512	1.3035	1.2218	1.1813	1.2593	1.2184	1.1437	1.1085	1.1802
11	1.5600	1.6659	1.5967	1.4975	1.4405	1.5338	1.4764	1.3882	1.3400	1.4237	1.3751	1.2954	1.2540	1.3301
12	1.7784	1.8990	1.8125	1.7013	1.6318	1.7352	1.6655	1.5686	1.5109	1.6020	1.5440	1.4578	1.4089	1.4907
13	2.0295	2.1727	2.0610	1.9318	1.8453	1.9634	1.8768	1.7828	1.6979	1.7992	1.7291	1.6342	1.5759	1.6652
14	2.3326	2.5152	2.3613	2.2079	2.0914	2.2319	2.1205	1.9937	1.9071	2.0229	1.9363	1.8294	1.7591	1.8585
15	2.7310	3.0002	2.7564	2.5423	2.3849	2.5685	2.4156	2.2599	2.1486	2.2868	2.1758	2.0511	1.9645	2.0783
16	3.3710		3.3911	3.0235	2.7809	3.0459	2.8045	2.5937	2.4413	2.6181	2.4661	2.3128	2.2020	2.3380
17					3.4107		3.4295	3.0675	2.6273	3.0885	2.8494	2.6416	2.4901	2.6644
18									3.4479		3.4656	3.1087	2.8707	3.1286
19													3.4829	

Table A.15 Compandors with Quantization Levels N = 2,...,36 for Signal with Laplacian Density (Mean = 0, Variance = 1)

LEVEL	2		3		4		5		6		7		8	
M.S.E.	.50008		.27703		.18346		.12608		.094180		.071622		.057136	
S.N.R.(DB)	3.0096		5.5749		7.3646		8.9936		10.260		11.450		12.431	
ENTROPY	1.0000		1.4719		1.8301		2.0831		2.3063		2.4880		2.6540	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.0000	.69827	.52809	0.00000	0.00000	.42089	.34952	0.00000	0.00000	.29888	.26113	0.00000	0.00000	.27191
2	1.7462		2.2956	1.2328	.94648	1.6468	1.2904	.76821	.64688	1.0634	.90558	.55898	.49232	.78923
3					2.6993		3.0232	1.9846	1.5822	2.2704	1.8358	1.3201	1.1346	1.5473
4									3.2956		3.5314	2.5185	2.0601	2.7379
5													3.7399	

LEVEL	9		10		11		12		13		14		15	
M.S.E.	.046056		.038296		.032058		.027429		.023579		.020599		.018059	
S.N.R.(DB)	13.367		14.169		14.941		15.618		16.275		16.861		17.433	
ENTROPY	2.7976		2.9308		3.0498		3.1614		3.2633		3.3595		3.4485	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	.20862	0.00000	0.00000	.18961	.17380	0.00000	0.00000	.16045	.14901	0.00000	0.00000	.13911	.13045	0.00000
2	.69977	.44001	.39785	.62878	.57102	.36313	.33404	.52310	.48268	.30929	.28799	.44811	.41821	.26945
3	1.3400	.99584	.88295	1.1833	1.0602	.80153	.73068	.96050	.87880	.67151	.62131	.80992	.75121	.57818
4	2.2614	1.7511	1.5264	1.9360	1.6970	1.3548	1.2190	1.5129	1.3663	1.1086	1.0170	1.2464	1.1464	.93962
5	3.9271	2.9348	2.4441	3.1135	2.6113	2.1053	1.8543	2.2614	2.0003	1.6597	1.5037	1.7966	1.6325	1.3756
6			4.0971		4.2530	3.2771	2.7655	3.4281	2.9087	2.4062	2.1365	2.5414	2.2641	1.9249
7							4.3970		4.5309	3.5684	3.0423	3.6993	3.1676	2.6681
8											4.6562		4.7738	3.8222

LEVEL	16		17		18		19		20		21		22	
M.S.E.	.016031		.014268		.012826		.011554		.010493		.0095446		.0087410	
S.N.R.(DB)	17.950		18.456		18.919		19.373		19.791		20.202		20.584	
ENTROPY	3.5332		3.6123		3.6879		3.7591		3.8274		3.8922		3.9545	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.12281	.11602	0.00000	0.00000	.10995	.10449	0.00000	0.00000	.099541	.095045	0.00000	0.00000	.090939
2	.25318	.39208	.36904	.23876	.22591	.34859	.33029	.21439	.20398	.31384	.29895	.19455	.18595	.28542
3	.54071	.70056	.65638	.50785	.47879	.61751	.58303	.45290	.42969	.55224	.52456	.40876	.38978	.49955
4	.87343	1.0617	.98883	.81610	.76594	.92554	.87000	.72167	.68230	.82086	.77705	.64705	.61530	.73774
5	1.2682	1.4971	1.3832	1.1769	1.0981	1.2860	1.2020	1.0295	.96911	1.1285	1.0637	.91554	.86768	1.0061
6	1.7537	2.0456	1.8681	1.6119	1.4922	1.7206	1.5957	1.3897	1.3009	1.4885	1.3952	1.2230	1.1542	1.3134
7	2.3843	2.7874	2.4977	2.1596	1.9765	2.2677	2.0795	1.8738	1.6943	1.9222	1.7884	1.5827	1.4855	1.6729
8	3.2856	3.9379	3.3971	2.9001	2.6052	3.0049	2.7074	2.3703	2.1776	2.4681	2.2713	2.0161	1.8784	2.1060
9	4.8847		4.9897	4.0473	3.5027	4.1510	3.6031	3.1081	2.8047	3.2050	2.8976	2.5614	2.3609	2.6508
10					5.0893		5.1842	4.2496	3.6989	4.3437	3.7903	3.2973	2.9866	3.3857
11									5.2747		5.3612	4.4335	3.8778	4.5196
12													5.4442	

Table A.15 (cont'd) Compandors with Quantization Levels N = 2,....,36 for Signal with Laplacian Density (Mean = 0, Variance = 1)

LEVEL	23		24		25		26		27		28		29	
M.S.E.	.0080160		.0073930		.0068263		.0063338		.0058825		.0054864		.0051212	
S.N.R.(DB)	20.960		21.312		21.658		21.983		22.304		22.607		22.906	
ENTROPY	4.0139		4.0712		4.1261		4.1791		4.2301		4.2794		4.3270	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	.087175	0.00000	0.00000	.083712	.080514	0.00000	0.00000	.077553	.074802	0.00000	0.00000	.072241	.069850	0.00000
2	.27307	.17809	.17086	.26175	.25133	.16420	.15805	.24172	.23282	.15234	.14703	.22455	.21685	.14208
3	.47683	.37251	.35670	.45611	.43712	.34219	.32883	.41966	.40356	.31647	.30501	.38865	.37480	.29436
4	.70227	.58656	.56040	.67009	.64076	.53760	.51457	.61392	.58925	.49437	.47571	.56651	.54547	.45842
5	.95454	.82466	.78576	.90810	.86604	.75041	.71815	.82776	.79277	.68857	.66136	.76066	.73108	.63625
6	1.2409	1.0929	1.0379	1.1762	1.1181	.98834	.94335	1.0656	1.0179	.90236	.86483	.97439	.93450	.83035
7	1.5721	1.4000	1.3242	1.4832	1.4042	1.2564	1.1953	1.3335	1.2698	1.1401	1.0899	1.2121	1.1595	1.0440
8	1.9647	1.7393	1.6552	1.8423	1.7351	1.5632	1.4814	1.6403	1.5557	1.4080	1.3417	1.4798	1.4115	1.2817
9	2.4468	2.1921	2.0475	2.2748	2.1272	1.9221	1.8120	1.9989	1.8862	1.7146	1.6275	1.7863	1.6970	1.5493
10	3.0718	2.7364	2.5293	2.8186	2.6066	2.3543	2.2039	2.4309	2.2779	2.0731	1.9579	2.1447	2.0272	1.8557
11	3.9617	3.4704	3.1537	3.5518	3.2324	2.8977	2.6850	2.9738	2.7587	2.5048	2.3494	2.5761	2.4185	2.2139
12	5.5239	4.6021	4.0423	4.6814	4.1198	3.6303	3.3083	3.7055	3.3814	3.0473	2.8299	3.1183	2.8987	2.6451
13			5.6005		5.6743	4.7578	4.1946	4.8313	4.2667	3.7782	3.4521	3.8485	3.5205	3.1869
14							5.7455		5.8142	4.9024	4.3364	4.9710	4.4038	3.9165
15											5.6807		5.9451	5.0374

LEVEL	30		31		32		33		34		35		36	
M.S.E.	.0047980		.0044984		.0042312		.0039824		.0037591		.0035502		.0033616	
S.N.R.(DB)	23.189		23.469		23.735		23.999		24.249		24.497		24.735	
ENTROPY	4.3732		4.4178		4.4612		4.5032		4.5441		4.5838		4.6225	
I	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)	X(I)	Y(I)
1	0.00000	.067612	.065514	0.00000	0.00000	.063543	.061687	0.00000	0.00000	.059937	.058283	0.00000	0.00000	.056719
2	.13745	.20967	.20294	.13312	.12905	.19664	.19072	.12522	.12162	.18514	.17989	.11821	.11500	.17492
3	.28443	.36192	.34990	.27515	.26646	.33865	.32811	.25830	.25063	.31821	.30888	.24340	.23659	.30010
4	.44235	.52595	.50779	.42738	.41339	.49085	.47502	.40629	.38800	.46018	.44624	.37645	.36557	.43313
5	.61299	.70374	.67839	.59138	.57126	.65483	.63286	.55247	.53489	.61233	.59311	.51840	.50291	.57506
6	.79855	.89780	.86393	.76923	.74183	.83255	.80340	.73642	.69271	.77626	.75090	.67054	.64976	.72718
7	1.0019	1.1114	1.0673	.96315	.92732	1.0265	.98886	.89411	.86322	.95391	.92139	.83444	.80753	.89105
8	1.2269	1.3490	1.2922	1.1767	1.1306	1.2401	1.1921	1.0880	1.0487	1.11478	1.1068	1.0121	.97799	1.0686
9	1.4786	1.6166	1.5438	1.4142	1.3555	1.4775	1.4169	1.3015	1.2519	1.3612	1.3099	1.2059	1.1634	1.2625
10	1.7642	1.9228	1.8293	1.6817	1.6070	1.7449	1.6684	1.5389	1.4766	1.5986	1.5346	1.4193	1.3665	1.4758
11	2.0943	2.2809	2.1593	1.9879	1.8926	2.0510	1.9537	1.8062	1.7280	1.8658	1.7860	1.6566	1.5911	1.7131
12	2.4854	2.7119	2.5502	2.3458	2.2223	2.4087	2.2833	2.1122	2.0133	2.1717	2.0712	1.9238	1.8424	1.9802
13	2.9654	3.2534	3.0299	2.7766	2.6131	2.8393	2.6742	2.4698	2.3429	2.5292	2.4008	2.2296	2.1275	2.2859
14	3.5867	3.9823	3.6508	3.3177	3.0926	3.3802	3.1534	2.9003	2.7335	2.9595	2.7911	2.5870	2.4570	2.6432
15	4.4690	5.1017	4.5323	4.0461	3.7130	4.1080	3.7734	3.4408	3.2124	3.4997	3.2699	3.0170	2.8473	3.0731
16	6.0075		6.0490	5.1641	4.5937	5.2246	4.6533	4.1681	3.8321	4.2265	3.8892	3.5570	3.3258	3.6128
17					6.1268		.61840	5.2835	4.7112	5.3406	4.7676	4.2833	3.9448	4.3386
18									6.2395		6.2936	5.3962	4.8224	5.4504
19													6.3463	

APPENDIX B

Derivation of Approximate Rantional Function Operator

The SDF of an NC1 model image is given by

$$S(z_1,z_2) = \frac{1 - r\alpha(z_1 + z_1^{-1} + z_2 + z_2^{-1})}{[1 - \alpha(z_1 + z_1^{-1} + z_2 + z_2^{-1})]^2} \quad . \qquad \text{(B-1)}$$

Let $R(z_1,z_2) \triangleq S(z_1,z_2)^{-\frac{1}{2}}$, thus

$$R(z_1,z_2) = \frac{1 - \alpha(z_1 + z_1^{-1} + z_2 + z_2^{-1})}{[1 - r\alpha(z_1 + z_1^{-1} + z_2 + z_2^{-1})]^{\frac{1}{2}}} \quad . \qquad \text{(B-2)}$$

We know from the following axiom that

$$(1-x)^{-n} = 1 + nx + \frac{n(n+1)}{2!}x^2 + \frac{n(n+1)(n+2)}{3!}x^3 + \ldots \qquad \text{(B-3)}$$

provided $x^2 < 1$ exists.

Then the alternative expression of $R(z_1,z_2)$ is, for simplicity, let $x \triangleq \alpha(z_1 + z_1^{-1} + z_2 + z_2^{-1})$,

$$R(z_1,z_2) = \frac{1-x}{(1-rx)^{\frac{1}{2}}}$$

$$= (1-x)\left[1 + \tfrac{1}{2}rx + \frac{1\cdot 3}{2^2}\frac{r^2x^2}{2!} + \ldots + \frac{1\cdot 3\cdot 5\cdots(2n-1)}{2n}\frac{r^nx^n}{n!} + \ldots\right]$$

$$= 1 + \tfrac{1}{2}rx + \frac{1\cdot 3}{2^2}\frac{r^2x^2}{2!} + \ldots + \frac{1\cdot 3\cdot 5\cdots(2n-1)}{2^n}\frac{r^nx^n}{n!} + \ldots$$

$$-x - \tfrac{1}{2}rx^2 - \frac{1\cdot 3}{2^2}\frac{r^2x^3}{2!} - \ldots - \frac{1\cdot 3\cdot 5\cdots(2n-3)}{2^{n-1}}\frac{r^{n-1}x^n}{(n-1)!}$$

$$- \frac{1\cdot 3\cdot 5\cdots(2n-1)}{2^n}\frac{r^nx^{n+1}}{n!} - \ldots$$

$$= 1 - \sum_{n=1}^{\infty} \frac{1\cdot 3\cdot 5\cdots(2n-3)}{2^{n-1}(n-1)!} \left[- \frac{(2n-1)r}{2n} + 1 \right] r^{n-1} x^n$$

$$= 1 - \sum_{n=1}^{\infty} \frac{1\cdot 3\cdot 5\cdots(2n-3)}{2^{n-1}(n-1)!} \left(1 - r + \frac{r}{2n} \right) r^{n-1} x^n \qquad \text{(B-4)}$$

Substitution of $n = 1,2,\ldots,$ into (B-4), and after some algebraic manipulations, different order of $R^{(n)}(z_1,z_2)$ is easily obtained as

$$R^{(1)}(z_1,z_2) = 1 - (1 - \tfrac{r}{2})\,\alpha\,(z_1 + z_1^{-1} + z_2 + z_2^{-1}) \qquad \text{(B-5)}$$

$$\begin{array}{ccc} & 0 & \\ 0 & \cdot & 0 \\ & 0 & \end{array} \qquad \begin{array}{l} \cdot = 1 \\ 0 = -(1-\tfrac{r}{2})\alpha \end{array}$$

Figure B.1 Construction Diagram of $R^{(1)}(z_1,z_2)$

$$R^{(2)}(z_1,z_2) = 1 - \tfrac{1}{2}(1 - r + \tfrac{r}{4})r4\alpha^2 - (1-\tfrac{r}{2})\,\alpha\,(z_1 + z_1^{-1} + z_2 + z_2^{-1})$$

$$-\tfrac{1}{2}(1 - r + \tfrac{r}{4})r\alpha^2[z_1^2+z_1^{-2}+z_2^2+z_2^{-2} + 2(z_1z_2+z_1z_2^{-1}+z_1^{-1}z_2+z_1^{-1}z_2^{-1})] \qquad \text{(B-5)}$$

$$\begin{array}{ccccc} & & x & & \\ & 2x & 0 & 2x & \\ x & 0 & \cdot & 0 & x \\ & 2x & 0 & 2x & \\ & & x & & \end{array} \qquad \begin{array}{l} \cdot = 1 - \tfrac{1}{2}(1 - r + \tfrac{r}{4})r4\alpha^2 \\ 0 = -(1-\tfrac{r}{2})\alpha \\ x = -\tfrac{1}{2}(1 - r + \tfrac{r}{4})r\alpha^2 \end{array}$$

Figure B.2 Construction Diagram of $R^{(2)}(z_1,z_2)$

$$
\begin{aligned}
R^{(3)}(z_1,z_2) = {} & 1 - \tfrac{1}{2}(1 - r + \tfrac{r}{4})r4\alpha^2 - \left[(1 - \tfrac{r}{2})\alpha + \frac{1\cdot 3}{2^2\cdot 2!}\,(1 - r + \tfrac{r}{6})r^2 9\alpha^3\right] \\
& \cdot(z_1+z_1^{-1}+z_2+z_2^{-1}) - \tfrac{1}{2}(1 - r + \tfrac{r}{4})r\alpha^2\Big[z_1^2+z_1^{-2}+z_2^2+z_2^{-2} \\
& + 2(z_1z_2+z_1z_2^{-1}+z_1^{-1}z_2+z_1^{-1}z_2^{-1})\Big] - \frac{1\cdot 3}{2^2\cdot 2!}\,(1 - r + \tfrac{r}{6})r^2\alpha^3 \\
& \cdot\Big[z_1^3+z_1^{-3}+z_2^3+z_2^{-3}+3(z_1^2z_2+z_1^2z_1^{-1}+z_1^{-2}z_2+z_1^{-2}z_2^{-1}+z_1z_2^2+z_1z_2^{-2} \\
& +z_1^{-1}z_2^2+z_1^{-1}z_2^{-2}\Big]
\end{aligned}
$$

$$
\begin{array}{ccccccc}
 & & & \Delta & & & \\
 & & 3\Delta & x & 3\Delta & & \\
 & 3\Delta & 2x & 0 & 2x & 3\Delta & \\
\Delta & x & 0 & \cdot & 0 & x & \Delta \\
 & 3\Delta & 2x & 0 & 2x & 3\Delta & \\
 & & 3\Delta & x & 3\Delta & & \\
 & & & \Delta & & &
\end{array}
\qquad
\begin{aligned}
\cdot &= 1 - \tfrac{1}{2}(1 - r + \tfrac{r}{4})r4\alpha^2 \\
0 &= -(1 - \tfrac{r}{2})\alpha - \frac{1\cdot 3}{2^2\cdot 2!}(1 - r + \tfrac{r}{6})r^2 9\alpha^3 \\
x &= -\tfrac{1}{2}(1 - r + \tfrac{r}{4})r\alpha^2 \\
\Delta &= -\frac{1\cdot 3}{2^2\cdot 2!}\,(1 - r + \tfrac{r}{6})r^2\alpha^3
\end{aligned}
$$

Figure B.3 Construction Diagram of $R^{(3)}(z_1,z_2)$

From the foregoing three Figures, $R^{(4)}(z_1,z_2)$ and $R^{(5)}(z_1,z_2)$ can be obtained in the similar manner

$$
\begin{array}{ccccccccc}
 & & & & \square & & & & \\
 & & & 4\square & \Delta & 4\square & & & \\
 & & 6\square & 3\Delta & x & 3\Delta & 6\square & & \\
 & 4\square & 3\Delta & x' & 0 & x' & 3\Delta & 4\square & \\
\square & \Delta & x & 0 & \cdot & 0 & x & \Delta & \square \\
 & 4\square & 3\Delta & x' & 0 & x' & 3\Delta & 4\square & \\
 & & 6\square & 3\Delta & x & 3\Delta & 6\square & & \\
 & & & 4\square & \Delta & 4\square & & & \\
 & & & & \square & & & &
\end{array}
$$

$$\cdot = 1 - \tfrac{1}{2}(1 - r + \tfrac{r}{4}) r 4\alpha^2 - \frac{1\cdot 3\cdot 5}{2^3\cdot 3!}\,(1 - r + \tfrac{r}{8}) r^3 36\alpha^4$$

$$0 = -(1 - \tfrac{r}{2})\alpha - \frac{1\cdot 3}{2^2\cdot 2!}\,(1 - r + \tfrac{r}{6}) r^2 9\alpha^3$$

$$x = -\left[\tfrac{1}{2}(1 - r + \tfrac{r}{4}) r\alpha^2 + \frac{1\cdot 3\cdot 5}{2^3\cdot 3!}\,(1 - r + \tfrac{r}{8}) r^3 16\alpha^4\right]$$

$$x' = -\left[(1 - r + \tfrac{r}{4}) r\alpha^2 + \frac{1\cdot 3\cdot 5}{2^3\cdot 3!}\,(1 - r + \tfrac{r}{8}) r^3 24\alpha^4\right]$$

$$\Delta = -\frac{1\cdot 3}{2^2\cdot 2!}\,(1 - r + \tfrac{r}{6}) r^2\alpha^3$$

$$\square = -\frac{1\cdot 3\cdot 5}{2^3\cdot 3!}\,(1 - r + \tfrac{r}{8}) r^3\alpha^4$$

Figure B.4 Construction Diagram of $R^{(4)}(z_1,z_2)$

$$
\begin{array}{ccccccccccc}
 & & & & & + & & & & & \\
 & & & & 5+ & \square & 5+ & & & & \\
 & & & 10+ & 4\square & \Delta & 4\square & 10+ & & & \\
 & & 10+ & 6\square & \Delta' & x & \Delta' & 6\square & 10+ & & \\
 & 5+ & 4\square & \Delta' & x' & 0 & x' & \Delta' & 4\square & 5+ & \\
+ & \square & \Delta & 0 & 0 & \cdot & 0 & x & \Delta' & \square & + \\
 & 5+ & 4\square & \Delta' & x' & 0 & x' & \Delta' & 4\square & 5+ & \\
 & & 10+ & 6\square & \Delta' & x & \Delta' & 6\square & 10+ & & \\
 & & & 10+ & 4\square & \Delta & 4\square & 10+ & & & \\
 & & & & 5+ & \square & 5+ & & & & \\
 & & & & & + & & & & &
\end{array}
$$

$$\cdot = 1 - \tfrac{1}{2}(1 - r + \tfrac{r}{4}) r 4\alpha^2 - \frac{1\cdot 3\cdot 5}{2^3\cdot 3!}(1 - r + \tfrac{r}{8}) r^3 36\alpha^4$$

$$0 = -\left[(1 - \tfrac{r}{2})\alpha + \frac{1\cdot 3}{2^2\cdot 2!}(1 - r + \tfrac{r}{6}) r^2 9\alpha^3 + \frac{1\cdot 3\cdot 5\cdot 7}{2^4\cdot 4!}(1 - r + \tfrac{r}{10}) r^4 100\alpha^5\right]$$

$$x = -\left[\tfrac{1}{2}(1 - r + \tfrac{r}{4}) r\alpha^2 + \frac{1\cdot 3\cdot 5}{2^3\cdot 3!}(1 - r + \tfrac{r}{8}) r^3 16\alpha^4\right]$$

$$x' = -\left[(1 - r + \tfrac{r}{4}) r\alpha^2 + \frac{1\cdot 3\cdot 5}{2^3\cdot 3!}(1 - r + \tfrac{r}{8}) r^3 24\alpha^4\right]$$

$$\Delta = -\left[\frac{1\cdot 3}{2^2\cdot 2!}(1 - r + \tfrac{r}{6}) r^2\alpha^3 + \frac{1\cdot 3\cdot 5\cdot 7}{2^4\cdot 4!}(1 - r + \tfrac{r}{10}) r^4 25\alpha^5\right]$$

$$\Delta' = -\left[3\,\frac{1\cdot 3}{2^2\cdot 2!}(1 - r + \tfrac{r}{6}) r^2\alpha^3 + \frac{1\cdot 3\cdot 5\cdot 7}{2^4\cdot 4!}(1 - r + \tfrac{r}{10}) r^4 80\alpha^5\right]$$

$$\square = -\frac{1\cdot 3\cdot 5}{2^3\cdot 3!}(1 - r + \tfrac{r}{8}) r^3\alpha^4$$

$$+ = -\frac{1\cdot 3\cdot 5\cdot 7}{2^4\cdot 4!}(1 - r + \tfrac{r}{10}) r^4\alpha^5$$

Figure B.5 Construction Diagram of $R^{(5)}(z_1,z_2)$

It is noted that when the order of $R^{(n)}(z_1,z_2)$ goes up, more sample points in the spatial domain would be involved in calculating the approximate rational function operator $\mathcal{L}$.

APPENDIX C

Unique Inverse Solution of Function $g_i'(x)$

The function $g_i'(x)$ is given by

$$g_i'(x) = \frac{(-2\log_e 2)2^{-2x}}{(1 - \rho_i^2 2^{-2x})^2} \tag{C-1}$$

Let $h(\cdot)$ be the inverse of $g'(\cdot)$. For convenience we denote $g_i'(x)$ and 2^{-2x} by z and y respectively, and rewrite (C-1) as

$$z = \frac{(-2\log_e 2)\cdot y}{1 - 2\rho_i^2 y + \rho_i^4 y^2} \tag{C-2}$$

Since $y = 2^{-2x}$, we have $1 \geq y \geq 0$ for any given $x \geq 0$. Rearranging (C-2) and getting an explicit expression as a function of y, we have

$$(\rho_i^4 z)y^2 - 2(\rho_i^2 z - \log_e 2)y + z = 0 \ . \tag{C-3}$$

The roots of (C-3) are

$$y_1 = \frac{-(-\rho_i^2 z + \log_e 2) + \sqrt{-2\rho_i^2 z \log_e 2 + (\log_e 2)^2}}{\rho_i^4 z} \tag{C-4a}$$

and

$$y_2 = \frac{-(-\rho_i^2 z + \log_e 2) - \sqrt{-2\rho_i^2 z \log_e 2 + (\log_e 2)^2}}{\rho_i^4 z} \ . \tag{C-4b}$$

The admissible root should lie in the interval [0,1]. Since $1 > \rho_i^2 > 0$, and from (C-1), we can observe that

$$0 \geq z \geq -\frac{2\log_e 2}{(1 - \rho_i^2)^2} \quad \text{for } \forall\, n \geq 0 \tag{C-5}$$

i.e., z is nonpositive. Therefore

$$\sqrt{-2\rho_i^2 z \log_e 2 + (\log_e 2)^2} > 0 \tag{C-6a}$$

$$-\rho_i^2 z + \log_e 2 > 0 \tag{C-6b}$$

and it is easy to prove that

$$-\rho_i^2 z + \log_e 2 > \sqrt{-2\rho_i^2 z \log_e 2 + (\log_e 2)^2} \; . \tag{C-7}$$

With the above conditions (C-6) and (C-7), we conclude

$$y_2 \geq y_1 \geq 0 \; . \tag{C-8}$$

In a similar manner, using (C-6) and (C-7), it is easy to check that

i) $y_2 > 1$, hence y_2 is inadmissible.

ii) $1 \geq y_1 \geq 0$, y_1 is the unique solution.

LIST OF REFERENCES

1. N. Ahmed, T. Natrarajan, and K.R. Rao, "On Image Processing and a Discrete Cosine Transform," IEEE Transactions on Computers, Vol. C-23, No. 1, pp. 90-93, January 1974.

2. V.R. Algazi, "Useful Approximations to Optimum Quantization," IEEE Transactions on Communication Technology, Vol. COM-14, No. 3, pp. 297-301, June 1966.

3. H.C. Andrews, Computer Techniques in Image Processing, Academic Press, New York, 1970.

4. H.C. Andrews, J. Kane, and W.K. Pratt, "Hadamard Transform Image Coding," Proceedings of the IEEE, Vol. 57, No. 1, January 1969.

5. H.C. Andrews and W.K. Pratt, "Fourier Transform Coding of Images," Hawaii International Conference on System Science, pp. 677-679, January 1968.

6. E. Angel and A.K. Jain, "Filtering of Multidimensional Diffusion Processes," 6th Asilomar Conference on Circuits and Systems, November 1972.

7. R. Bellman, Introduction to Matrix Analysis, McGraw Hill, New York, 1970.

8. P. Castellino, G. Madena, L. Nebbia, and C. Scngliala, "Bit Rate Reduction by Automatic Adaptation of Quantizer Step-Size in DPCM Systems," in Proc. 1974 Int. Zurich Seminar Digital Communications, pp. B6(1)-B6(6), 1974.

9. C.C. Cutler, "Differential Quantization of Communication Signals," U.S. Patent No. 2,605,301, July 29, 1952.

10. J.E. Essman, "Theory and Application of DPCM and Compression of DPCM with other Time Predictive Techniques," Ph.D. Thesis, Purdue Univ., Electrical Engineering, 1972.

11. B. Fox, "Discrete Optimization via Marginal Analysis," Management Science, Vol. 13, pp. 201-216, November 1966.

12. L.E. Franks, "A Model for the Random Video Process," Bell Systems Technical Journal, pp. 609-630, April 1966.

13. W. Frei and C.C. Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," IEEE Transactions on Computers, Vol. C-26, No. 10, pp. 988-998, October 1977.

14. J.I. Gimlett, "Use of Activity Classes in Adaptive Transform Image Coding," IEEE Transactions on Communications, Correspondence, Vol. 23, pp. 785-786, July 1975.

15. D.N. Graham, "Image Transmission by Two-Dimensional Contour Coding," Proceedings of the IEEE, Vol. 55, No. 3, pp. 336-346, March 1967.

16. A.K. Griffith, "Edge Detection in Simple Scenes Using Apriori Information," IEEE Transactions on Computers, Vol. C-22, pp. 371-381, April 1973.

17. A. Habibi, "Hybrid Coding of Pictorial Data," IEEE Transactions on Communications, Vol. COM-22, No. 5, pp. 614-624, May 1974.

18. A. Habibi, "Study of On-Board Compression of Earth Resources Data," Final Report, Contract No. NAS 2-8394, TRW No. 26566, September 1975.

19. A. Habibi, "Survey of Adaptive Image Coding Techniques," IEEE Transactions on Communications, Vol. COM-25, No. 11, pp. 1275-1284, November 1977.

20. A. Habibi, "Two Dimensional Bayesian Estimate of Images," Proceedings of the IEEE, Vol. 60, pp. 878-883, July 1972.

21. A. Habibi and P.A. Wintz, "Image Coding Linear Transformation and Block Quantization," IEEE Transactions on Communication Technology, Vol. COM-19, No. 1, pp. 50-62, February 1971.

22. A. Habibi and P.A. Wintz, "Linear Transformation for Encoding Two-Dimensional Sources," School of Electrical Engineering, Purdue Univ., Lafayette, Indiana, Technical Report EE 70-2, March 1970.

23. C.W. Harrison, "Experiments with Linear Prediction in Television," The Bell Systems Technical Journal, Vol. 31, No. 4, pp. 746-783, July 1952.

24. M.H. Hueckel, "An Operator Which Locates Edges in Digitized Pictures," J.A.C.M., Vol. 18, January 1971.

25. B.R. Hunt, "Digital Image Processing," Proceedings of the IEEE, Vol. 63, No. 4, pp. 693-708, April 1975.

26. A.K. Jain, "A Fast Karhunen-Loeve Transform for a Class of Random Processes," IEEE Transactions on Communications, Concise Paper, pp. 1023-1029, September 1976.

27. A.K. Jain, "A Fast Karhunen-Loeve Transform for Finite Discrete Images," Proceedings of the National Electronics Conference, Chicago, Illinois, pp. 323-328, October 1974.

28. A.K. Jain, "A Semicausal Model for Recursive Filtering of Two-Dimensional Images," IEEE Transactions on Computers, Vol. C-26, No. 4, pp. 343-350, April 1977.

29. A.K. Jain, "A Sinusoidal Family of Unitary Transforms," Technical Report, Dept. of Electrical Engineering, SUNY at Buffalo, New York, July 1977.

30. A.K. Jain, "Image Coding via a Nearest Neighbors Image Model," IEEE Transactions on Communications, Vol. COM-23, No. 3, pp. 318-331, March 1975.

31. A.K. Jain, "Partial Differential Equations and Finite-Difference Methods in Image Processing, Part I: Image Representation," Journal of Optimization Theory and Applications, Vol. 23, No. 1, pp. 65-91, September 1977.

32. A.K. Jain, "Some New Techniques in Image Processing," Image Science Math., Proceedings ONR Symposium on Current Problems in Image Science (1976), (C.O. Wilde & E. Barrett, Editors), West. Periodicals Co., No.Hollywood, California, 1977.

33. A.K. Jain and E. Angel, "Image Restoration, Modeling, and Reduction of Dimensionality," IEEE Transactions on Computers, Vol. C-23, No. 5, pp. 470-476, May 1974.

34. A.K. Jain and J.R. Jain, "Partial Differential Equations and Finite Difference Methods in Image Processing, Part II: Image Restoration," IEEE Transactions on Automatic Control, Vol. AC-26, No. 5, pp. 817-834, October 1978.

35. A.K. Jain and S.H. Wang, "Stochastic Image Models and Hybrid Coding," Naval Ocean Systems Center Contract NOO 953-77-C-003 MJE, Final Report, Dept. of Electrical Engineering, SUNY at Buffalo, February 1978.

36. A.K. Jain, S.H. Wang and Y.Z. Liao, "Fast Karhunen Loeve Transform Data Compression Studies," 1976 National Telecommunications Conference, Dallas, Texas, December 1976.

37. H. Karhunen, "On Linear Methods in Probability Theory," (1947) (English translation by I. Selin), The Rand Corporation, Doc. T-131, August 11, 1960.

38. R.L. Kashyap and A.R. Rao, Dynamic Stochastic Models from Empirical Data. Academic Press, New York, 1976.

39. R. Kirsch, "Computer Determination of the Constituent Structure of Biological Images," Comput. Biomed. Res., Vol. 4, pp. 315-328, 1971.

40. A.J. Kurtenbach and P.A. Wintz, "Optimum Quantization," Supplement to IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-3, pp. 563-580, November 1967.

41. M. Levine, "Feature Extraction: A Survey," Proceedings of the IEEE, Vol. 57, pp. 1391-1405, August 1969.

42. M. Loeve, Fonctions Aléatories de Seconde Ordre, Hermann, Paris, 1948.

43. J.L. Mannos and D.J. Sakrison, "The Effects of a Visual Fidelity Criterion on the Encoding of Images," IEEE Transactions on Information Theory, Vol. IT-20, No. 4, pp. 525-536, July 1974.

44. J. Max, "Quantizing for Minimum Distortion," IRE Transactions on Information Theory, Vol. IT-6, pp. 7-12, March 1960.

45. R.A. McDonald, "Signal-to-Noise and Idle Channel Performance of DPCM Systems - Particular Applications to Voice Signals," B.S.T.J., Vol. 45, No. 7, pp. 1123-1151, September 1966.

46. R.W. Means, E.H. Wrench and J.J. Whitehouse, "Image Transmission via Spread Spectrum Techniques," ARPA Quarterly Technical Report, ARPA-QR6, Naval Undersea Center, San Diego, California 92132, Jan.-Dec. 1975.

47. A.Z. Meiri, "The Pinned Karhunen-Loeve Transform of a Two Dimensional Gauss-Markov Field," Proceedings 1976 SPIE Conference on Image Processing, San Diego, California, 1976.

48. N.E. Nahi and T. Assefi "Bayesian Recursive Image Enhancement," IEEE Transactions on Computers, Vol. C-12, pp. 734-738, July 1972.

49. N.E. Nahi and C.A. Franco, "Recursive Image Enhancement-Vector Processing," IEEE Transactions on Communications, Vol. COM-21, pp.305-311, April 1973.

50. J.B. O'Neal, "Predictive Quantization Systems (Differential Pulse Code Modulation) for the Transmission of Television Signals," The Bell Systems Technical Journal, Vol. 45, No. 5, pp. 689-721, May-June 1966.

51. M.D. Paez and T.H. Glisson, "Minimum Mean-Squared-Error Quantization in Speech PCM and DPCM Systems," IEEE Transactions on Communications (Concise Papers), Vol. COM-20, pp. 225-230, April 1972.

52. W.K. Pratt, "Bibliography on Digital Image Processing and Related Topics," Electronics Sciences Laboratory, University of Southern California, USCEE Report 453, September 1973.

53. W.K. Pratt, Digital Image Processing, John Wiley & Sons Inc., 1978.

54. W.K. Pratt, H.C. Andrews, and J. Kane, "Hadamard Transform Image Coding," Proceedings of the IEEE, Vol. 57, No. 1, pp. 58-68, January 1969.

55. W.K. Pratt, W.H. Chen and L.R. Welch, "Slant Transform Image Coding," IEEE Transactions on Communicatons, Vol. COM-22, No. 8, pp. 1075-1093, August 1974.

56. W.D. Ray and R.M. Driver, "Further Decomposition of the Karhunen-Loeve Series Representation of a Stationary Random Process," IEEE Trans. Information Theory, Vol. IT-16, No. 6, pp. 663-668, November 1970.

57. L.G. Robert, "Machine Perception of Three Dimensional Solids," in Optical and Electro-Optical Information Processing, J.T. Tippett et al., Eds., MIT Press, Cambridge, MA, pp. 187-197, 1965.

58. J.A. Roese, W.K. Pratt, and G.S. Robinson, "Interframe Cosine Transform Image Coding," IEEE Transactions on Communications, Vol. COM-25, pp. 1329-1339, November 1977.

59. R.P. Roesser, "A Discrete State-Space Model for Linear Image Processing," IEEE Transactions on Automatic Control, Vol. AC-20, No. 1, pp. 1-10, February 1975.

60. A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEE Transactions on Computers, Vol. C-20, pp. 562-569, May 1971.

61. A.P. Sage and J.L. Melsa, Estimation Theory with Applications to Communications and Control, McGraw-Hill, 1971.

62. W.F. Schreiber, "Picture Coding," Proceedings of the IEEE, Vol. 55, No. 3, pp. 320-335, March 1967.

63. W.F. Schreiber, T.S. Huang, and O.J Tretiak, "Contour Coding of Images," IRE Western Electron Show and Conv., pp. 8/3(1)-8/3(6), August 1968.

64. W.F. Schreiber and C.F. Knapp, "TV Bandwidth Reduction by Digital Coding," IRE Natinal Convention Record, Vol. 6, Part 4, pp. 88-89, 1958.

65. W.F. Schreiber, C.F. Knapp, and N.D. Kay, "Synthetic Highs: An Experimental TV Bandwidth Reduction System," J. Soc. Motion Picture and Television Engineers, Vol. 68, pp. 525-537, August 1959.

66. A. Segall, "Bit Allocation and Encoding for Vector Sources," IEEE Transactions on Information Theory, Vol. IT-22, No. 2, pp. 162-169, March 1976.

67. C.E. Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion," 1959 IRE Nat'l Conv. Rec., Pt. 4, pp. 152-163.

68. R. W. Stroh, "Optimum and Adaptive Differential Pulse Code Modulation," Ph.D. Thesis, Polytechnic Institute of Brooklyn, Electrical Engineering, 1970.

69. L.C. Wilkins and P.A. Wintz, "Bibliography on Data Compression, Picture Properties, and Picture Coding," IEEE Transactions on Information Theory, Vol. IT-17, No. 2, pp. 180-197, March 1971.

70. P.A. Wintz, "Transform Picture Coding," Proceedings of the IEEE, Vol. 60, No. 7, pp. 809-820, July 1972.

71. E. Wong, "Two Dimensional Random Fields and Representation of Images," SIAM J. Appl. Math., Vol. 16, pp. 756-770, July 1968.

72. R.C. Wood, "On Optimum Quantization," IEEE Transactions on Information Theory, Vol. IT-5, pp. 248-252, March 1969.

73. J.W. Woods, "Two Dimensional Discrete Markov Fields," IEEE Transactions on Information Theory, Vol. IT-18, pp. 232-240, March 1972.

74. J.K. Yan and D.J. Sakrison," Encoding of Images Based on a Two-Component Source Model," IEEE Transactions on Communications, Vol. COM-25, No. 11, pp. 1315-1322, November 1977.

75. "Special Issue on Redundancy Reduction," Proceedings of the IEEE, Vol. 55, No. 3, pp. 251-480, March 1967.

VITA

Shenq-Huey Wang was born in Shanghai, China, on April 8, 1948. He received the B.S. degree in Electrical Communication Engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 1970. He served as a Radio Communication Officer in the Chinese Army from 1970 to 1971. In August, 1971, he was employed as an engineer for General Instrument Corporation of Taiwan Ltd. During 1972-74 he was a graduate assistant with the Department of Electrical Engineering, University of Connecticut, and obtained his M.S.E.E. degree in May, 1974. In September, 1974, he joined the State University of New York at Buffalo, where he was a research and teaching assistant throughout his doctoral program. During the summer of 1975, he worked at NASA, Huntsville, Alabama, where he was involved in developing computer software for satellite image processing. He spent the last half year of his graduate study in the Department of Electrical Engineering, University of California at Davis, where his thesis supervisor was on sabbatical, to complete work on his Ph.D. thesis.

Presently, he holds the position of Image Scientist at the Picker Corporation, Cleveland, Ohio, where he is working on Computed Tomography. His research interests include Digital Signal and Image Processing, Data Compresssion and Modeling of Images, and Computer Graphics.